DIGITAL CHAOTIC COMMUNICATIONS

A Dissertation
Presented to
The Academic Faculty

By

Alan J. Michaels

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in ECE

Georgia Institute of Technology

August, 2009

DIGITAL CHAOTIC COMMUNICATIONS

Approved by:

Dr. Thomas D. Morley, Advisor
School of Mathematics
Georgia Institute of Technology

Dr. W. Marshall Leach, Co-Advisor
School of Electrical and
Computer Engineering
Georgia Institute of Technology

Dr. Mary Ann Ingram
School of Electrical and
Computer Engineering
Georgia Institute of Technology

Dr. David Citrin
School of Electrical and
Computer Engineering
Georgia Institute of Technology

Dr. John Dorsey
School of Electrical and
Computer Engineering
Georgia Institute of Technology

Dr. David B. Chester
Harris Corporation

Date Approved: 1 July 2009

Dr. Evans Harrell
School of Mathematics
Georgia Institute of Technology

Dedication

soli Deo Gloria

Acknowledgements

I am thankful to many sources that have contributed to this work, from direct advisement on the research, to financial support, to personal encouragement. First, Ashley, thank you for supporting the many absent nights, weekends, and overtime to dabble in chaos, sacrificing to provide me with encouragement, hot food, and clean underwear. Dave Chester and Tom Wells, thank you for your insightful mentorships and friendships: you have both helped me gain maturity in applying my skills. Professors Tom Morley and Marshall Leach, thank you for a firm academic foundation and patience in the PhD advisement process. And, finally the Harris Corporation, thank you for both the opportunity to explore this research and the personal financial investment in the form of Georgia Tech and Carnegie Mellon tuition.

Disclaimers:

1. The commercial application of the technology presented in this document is the subject of more than 30 pending patent applications (Appendix B) by the Harris Corporation.

2. The completion of the hardware prototype discussed in Chapter 3 of this dissertation represents a collective effort of the Harris Chaotic Comms IR&D team, converting the core analysis and simulations developed under the advisement of Harris Sr. Scientist, David Chester, into a measurable software defined radio. Those collective efforts are captured in sections 3.1, 3.2.1, 3.2.5, 3.2.6, and 3.3.6. In particular, thank you to David Browning, Dan Boritzki, Rory Fagan, Nick Miller, Joe Petrone, and Ravi Varanasi.

Table of Contents

List of Tables

List of Figures

Acronym	Reference	Acronym	Reference
ADC	Analog-to-Digital Converter	AES	Advanced Encryption Standard
AGC	Automatic Gain Control	APSK	Asymmetric Phase Shift Keying
ARIMA	Autoregressive Integrated Moving Average	ARM	Advanced Risc Machine
ASIC	Application Specific Integrated Circuit	ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise	BER	Bit Error Rate
BOC	Binary Offset Coding	BPSK	Binary Phase Shift Keying
BSS	Blind Source Separation	C/A code	Civilian Acquisition Code
CAZAC	Constant Amplitude Zero Autocorrelation	CDMA	Code Division Multiple Access
CLT	Central Limit Theorem	CMAC	Complex Multiply Accumulator
CMOS	Complementary Metal Oxide Semiconductor	COOK	Chaotic On-Off Keying
CORDIC	Coordinate Rotation Digital Computer	CPSK	Chaotic Phase Shift Keying
CRT	Chinese Remainder Theorem	CSD	Canonic Signed Digit
CW	Continuous Wave	DAC	Digital-to-Analog Converter
DCM	Digital Clock Manager	DCSK	Differential Chaotic Shift Keying
DDC	Digital Downconverter	DES	Data Encryption Standard
DFT	Discrete Fourier Transform	DS	Direct Sequence
DSP	Digital Signal Processing	FEC	Forward Error Correction
FFT	Fast Fourier Transform	FH	Frequency Hopping
FIFO	First-in First-out	FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array	GF	Galois Field
GLONASS	Global Navigation Satellite System	GPS	Global Positioning System
HPA	High Power Amplifier	IF	Intermediate Frequency
iid	Independent and Identically Distributed	IIR	Infinite Impulse Response
IR&D	Independent Research and Development	I/Q	In-phase and Quadrature-phase
ISI	Intersymbol Interference	LSB	Least Significant Bit
LUR	Loop Update Rate	LUT	Lookup Table
MCM	Multi-chip Module	MG	Masking Generator
MSB	Most Significant Bit	NCO	Numerically Controlled Oscillator
NLP	Nonlinear Processor	ODE	Ordinary Differential Equation
OFDM	Orthogonal Frequency Division Multiplexing	OOK	On-Off Keying
OTA	Over-the-Air	PAM	Pulse Amplitude Modulation
PAPR	Peak to Average Power Ratio	P code	Precise Code
PM	Phase Modulation	ppm	Parts per Million
PPS	Pulse Per Second	PRNG	Pseudorandom Number Generator
PSK	Phase Shift Keying	QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying	RADAR	Radio Detection and Ranging
RAM	Random Access Memory	RF	Radio Frequency
RG	Ring Generator	RMS	Root-Mean-Square
RNS	Residue Number System	ROM	Read-Only Memory
SAW	Surface Acoustic Wave	SDR	Software Defined Radio
SFDR	Spurious Free Dynamic Range	SiP	System-in-a-Package
SNR	Signal-to-Noise Ratio	SWaP	Size, Weight, and Power
TDMA	Time Division Multiple Access	TRL	Technical Readiness Level
USPTO	United States Patent and Trademark Office	UTC	Coordinated Universal Time
VHDL	VHSIC Hardware Description Language	WAAS	Wide Area Augmentation System
XOR	Exclusive OR		

Summary

This dissertation provides the conceptual development, modeling and simulation, physical implementation, and measured hardware results for a practicable digital coherent chaotic communication system. Such systems are highly desirable for robust communications due to the maximal entropy signal characteristics that satisfy Shannon's ideal noise-like waveform and provide optimal data transmission across a flat communications channel. At the core of the coherent chaotic communications system is a fully digital chaotic circuit, providing an efficiently controllable mechanism that overcomes the traditional bottleneck of chaotic circuit state synchronization. The analytical, simulation, and hardware results yield a generalization of direct sequence spread spectrum waveforms, that can be further extended to create a new class of maximal entropy waveforms suitable for optimized channel performance, maximal entropy transmission of chaotically spread amplitude modulated data constellations, and permission-based multiple access systems.

Chapter 1: Introduction to Chaotic Communication Systems

The field of chaotic communications has gone through various periods of intense interest, initiated by Shannon's 1947 recognition that the channel capacity of a communications link is optimized when the waveform is a noise-like maximal entropy signal[1] and further solidified by Chua's 1980 implementation of a practical chaotic elctrical circuit[2]. Chaotic communication systems resemble direct sequence spread spectrum communication systems in that the data is spread across a relatively wide transmission bandwidth and then despread by the intended receiver with a time-synchronized spreading sequence. These systems tend to be more computationally complex than non-spread communication systems, yet they provide advantageous multipath mitigation and multi-user spectral re-use capabilities. The chaotic sequence based communication systems exhibit analytically better performance than direct sequence based communication systems and may in general be viewed as a generalization of direct sequence approaches; as its layman's connotation would suggest, a chaotic sequence or system evolves in a seemingly random fashion, while the direct sequence system is limited to a small finite set of values. The most significant limitation of chaotic communication systems is the extreme precision needed to accurately synchronize and track two independent instantiations of an "identical" chaotic circuit as used at a transmitter and a receiver. Active research in chaotic communications was revived in the early 1990s when various chaotic circuit synchronization methods were demonstrated, leading to limited communications capabilities; each of the proposed methods had drawbacks that limited their practical implementations. During this period, the theoretical performance of chaotic communications has been shown by various authors[3, 4, 5, 6] to exceed that of direct sequence systems and, ultimately, to satisfy Shannon's noise-like waveform characteristics for a maximal entropy waveform that miximizes channel capacity. To date, nobody has demonstrated a sufficiently robust chaotic circuit synchronization method that supports practical coherent chaotic communications on par with direct sequence approaches.

This dissertation presents a divergence from the traditional chaotic communications approaches that harness analog chaotic circuits by implementing a fully digital chaotic circuit that is then conditioned for use in a prototype coherent chaotic communications system. Topics include a comparison of analog and digital chaotic circuits; implementation of digital chaotic circuits for use in chaotic communications; analytical, simulation, and measured hardware results for a prototype coherent communication system; and generalization of the fundamental chaotic waveform to multipath mitigation techniques, multiple access communication systems,

permission-based communication systems, and a new class of maximal entropy amplitude modulated chaotic waveform hybrids for use in specific applications.

1.1 History of Chaotic Communication Systems

The fundamental technology for coherent chaotic communications, similar to spread spectrum communications, grew out of harnessing FM radio non-idealities and RADAR technology[7]. These systems exploit the fact that an analog communications waveform can be created/detected as well as used to directly sense environmental characteristics, leading to correlation based receivers. In 1925, British scientists E.V. Appleton and M.A.F. Barnett observed that electric radiation bounced off of the ionized gas layer in the Earth's upper atmosphere[8]. This observation led to the development of FM altimetry in the 1930s and the conceptual leap of generalized statistical signal detection via RADAR. These techniques evolved into the early stages of spread spectrum technology when the Allied forces implemented SIGSALY (also known as the Green Hornet) on 15 July 1943[9]. SIGSALY was a Bell Telephone Laboratories (BTL) invention that grew out of digitizing voice signals (a vocoder) and provided secure spread spectrum communications between the Americans and British by modulating voice signals prior to transmission with recorded galactic noise stored on phonographs.[1] The voice signal was then demodulated on the other side of the Atlantic using a precisely time-synchronized replica of the original encoding sequence. Although six decades removed, SIGSALY is surprisingly similar to the instantiation of a modern analog chaotic waveform in that a seemingly random process is coupled with intelligible data.

In 1947, Claude Shannon of BTL showed[1] that the channel capacity in the presence of additive Gaussian noise is maximized by selective spreading of the transmitted signal over a bandwidth such that the sum total of signal power (a constant) and noise within the designated bandwidth are as uniformly low as possible. In other words, the signal is spread over a bandwidth such that the average power is minimized. This capacity limit is met when the signal is drawn from a set of noise-like waveforms that are received using matched filter/correlation techniques. Shannon expanded these results to communications through a noisy channel[10] as well as capacity measures for secure communications[11]. Building on BTL's mathematical foundation of channel capacity, five distinct classes of spread-spectrum technology evolved[7]: the robust yet cumbersome pure-noise solution (e.g. SIGSALY); direct sequence (DS) systems employing pseudorandom numbers that spread a carrier signal via phase-shift keying; frequency modulation with frequency wobbled over a wide bandwidth (e.g. chirped carrier

[1]Each phonograph contained 12 minutes worth of noise (effectively AWGN) and was duplicated so that coherent reception could occur on the other side of the Atlantic. Spreading sequence synchronization was performed using precision phonograph players that spun the discs at the same rate.

frequency); frequency hopping (FH) systems that employ a pseudorandom sequence to control a frequency synthesizer; and time hopping systems that randomize temporal emissions. These systems have been developed into numerous applications (e.g. RADAR, noise wheels, cellular telephony[12], and defense communications); a quick search of the US Patent Office shows over 17000 hits for the key words "spread spectrum." The chaotic communications focus of this dissertation is a digital generalization of direct sequence spread spectrum communications that functionally resembles the maximal entropy waveform characteristics of pure-noise systems.

1.2 Motivation for Chaotic Communication Systems

Spread-spectrum signals are well known to be resistant to interferers (natural and man-made) and multipath effects, conducive to secure communications by lowering the average spectral density, and effective for use in multiple access systems where users simultaneously re-use the shared communications bandwidth[13, 4]. A notional depiction of the spectral power density of a modulated data signal both before and after spreading is shown in Figure 1.

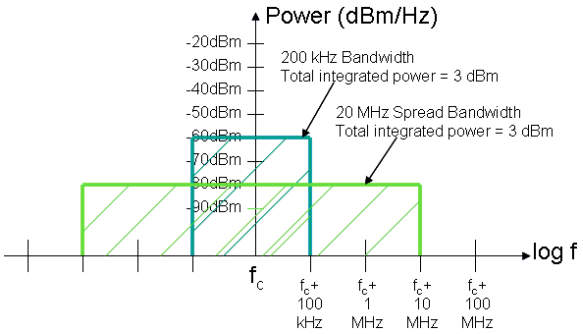

FIGURE 1. Power spectral density effects of signal spreading.

The extension of spread-spectrum signaling techniques to chaotic communications gained active interest in the early 1990s[14, 15, 3, 16] since frequency bandlimited chaotic spreading sequences are known to closely mimic Shannon's ideal noise-like waveform[1]; the chaotic waveform is a near-optimal approximation of a transmission with maximum capacity for carrying information in a Gaussian white noise channel. Compared to other spread communication systems, chaotic waveforms may be viewed as having the potential for higher throughputs (as a result of higher SNR) or a lower power spectral density (increasing spectral re-use) for the same data throughput. Further, the impulsive autocorrelation also gives chaotic waveforms superior multipath and co-interference characteristics as compared to traditional spread-spectrum signals like CDMA.

By contrast with a conventional digital modulation scheme, where the transmitted symbols are mapped to a finite set of periodic waveform segments for transmission, every transmitted symbol in a chaotic modulation scheme produces a different nonperiodic waveform segment. Because the cross correlations between pieces of periodic segments are lower than between pieces of periodic waveforms, chaotic modulation ought to offer better performance under multipath propagation conditions. Thus, chaotic modulation offers a potentially simple solution for robust wideband communications. [5]

The fundamental difference between a traditional direct sequence spread-spectrum communication system and a coherent chaotic sequence spread spectrum communication system is the absence of apparent periodicity in the chaotic waveform. The chaotic sequence is effectively a quadrature pair of independent Gaussian random variables as opposed to a (possibly pulse-shaped) string of constant-amplitude square-wave pulses. In general, any correlation, definable characteristic, or waveform feature can be viewed as lowering the entropy of the signal, moving away from Shannon's ideal noise-like waveform. As an example, consider the time-domain spreading sequences shown in Figure 2, where a four times oversampled chaotic sequence (blue) is plotted next to a comparable four times oversampled pulse-shaped DS spreading sequence (green).[2]

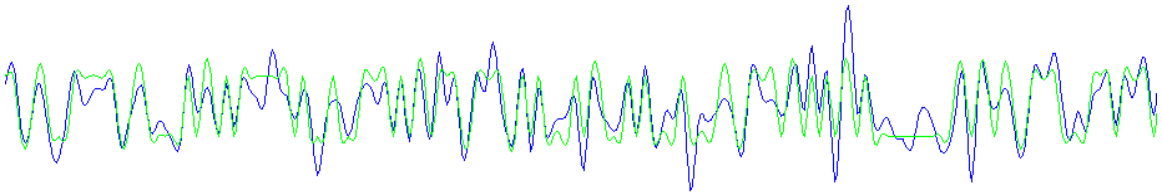

FIGURE 2. Comparison of chaotic (blue) and traditional DS (green) spreading sequences.

The combination of a quadrature pair of the chaotic spreading sequences will result in a uniformly distributed phase for the chaotic spreading sequence as opposed to a non-random cyclo-stationary distribution for the DS spreading sequence. These characteristics are shown notionally as histograms for a sample spreading sequence of length 1,000,000 in Figure 3.

[2]A representative pair of sequences may be constructed in Matlab using the commands:
a = randn(1,10000); b = interp(a,4); c = interp(sign(a),4);

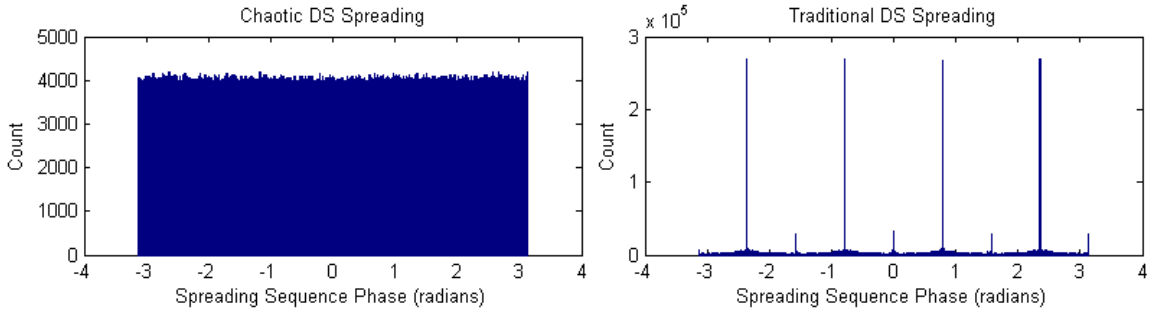

FIGURE 3. Comparison of phase histograms for chaotic (left) and traditional DS spreading (right).

Transitioning from a time-domain view to the frequency domain or a statistical analysis yields other verifications that the chaotic sequence modulated waveforms approximate the maximal entropy noise-like signals that Shannon described as optimal for transmission through AWGN channels. Chaotic waveforms have ideal flat spectral power densities and Gaussian distributed amplitudes on each of their in-phase and quadrature components, compared to peak-ish uniform amplitude distributions at four distinct phases for the traditional DS spreading waveform. These characteristics will be discussed and quantified thoroughly in this dissertation by the use of time domain, frequency domain, and statistical measures.

1.2.1 Fundamentals of Mathematical Chaos

Chaos is a summary categorization for dynamical systems that demonstrate perfectly predictable behavior, yet appear to be wildly amorphous and without meaningful order. The popular phrase "order from chaos" suggests violations of the second law of thermodynamics; quite contrary, chaos is already perfectly ordered, yet we typically lack the necessary understanding of the chaotic system to predict its future states. Given perfect measurements of the current state and the nonlinear dynamics that underlie the system, we can perfectly predict how the chaotic system will evolve. More practically, the higher precision that we can obtain in estimating the chaotic system state and its properties, the further into the future that we can bound its behavior.

A chaotic system is defined by three fundamental tenets[17]:

1. A chaotic system has a dense collection of points with periodic orbits
2. A chaotic system is extremely sensitive to initial conditions and perturbations
3. A chaotic system is topologically transitive

These definitions represent a precise mathematical view of the world, and within it chaotic systems, where truly continuous amplitudes can exist. Notionally, a chaotic system operates on some open continuous domain and takes every possible value within an open continuous

range (denseness)[18]; is periodic with potentially non-measurable period[19]; responds in a divergent manner to the smallest of perturbations (extreme sensitivity); and recursively operates on any value in the domain to produce a value that is arbitrarily close to any chosen value in the range (topological transitivity). Chaotic systems take numerous forms, with most being described as nonlinear differential equations; descriptors like attractors, bifurcations, and Lyuponov exponents are used to evaluate or measure the systems. The famous example of Lorenz' attractor is shown in Figure 4, with red and blue curves representing near identical initial conditions into the same chaotic system.

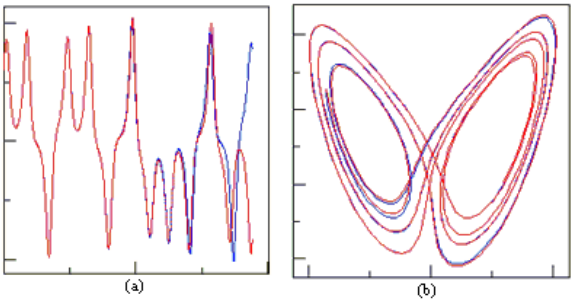

FIGURE 4. Plots of Lorenz' attractor in time domain (a) and phase space (b). The red and blue curves quickly diverge due to the difference in initial conditions.

These concepts hold perfectly in a universe where the domain and range values of the chaotic operator are continuous; in an engineering realm, however, most controllable parameters are discrete. Electrical circuits are fundamentally discrete as proven by Millikan's charged oil drop experiments[20], resulting in rational ($\mathbb{Q}$) values for circuit parameters like voltages, currents, resistances, etc. At a lower level, all matter is composed of discrete atoms, with quantized masses and valence electron energy levels. These observations by no means represent a pessimistic or dismissive view of the existing results in nonlinear dynamics or chaotic systems, but they do provide an impetus to consider fully discrete approximations to chaotic systems when engineering a chaotic communications system; e.g. implement the behavioral characteristics of a mathematically chaotic system in a robust engineering approximation rather than attempt to build an engineering model that integrates and depends upon chaotic systems.

Since all of the fundamental building blocks for implementing a communication system using a chaotic waveform are discrete, a logical step is to derive digital chaotic systems that behave chaotically. In particular, the conversion from differential equations to difference equations, i.e. from a continuous view of nonlinear dynamics to a discrete one, provides concepts for discrete-time chaotic mapping rules. Numerous discrete-time mappings have been proven to

exhibit all three properties of a chaotic system, from chaotic tent maps[6] to recursive polynomial evaluations[21, 22]. Since the value will have a finite precision in any physical system, we gravitate further into the realm of discrete-time discrete-amplitude chaotic polynomial mappings. The third-order Chebyshev polynomial, $T_3(x) = 4x^3 - 3x$[23], and the logistic equation $f(x) = x(1-x)$ [24] have both been shown to generate chaotic sequences when operating on finite sets of size 2^k scaled to $[0, 1)$. More efficient results using chaotic polynomial evaluations over residue number systems (RNS) coupled via the Chinese Remainder Theorem (CRT)[25] have also been demonstrated. The proposal preceding this dissertation described the hardware construction of a discrete-time discrete-amplitude pseudorandom number generation mechanism with chaotic properties. That approach, coupled with sequence generation extensions that improve the sequence length, form the basis for the prototype coherent chaotic communication system developed during this doctoral research.

1.2.2 Historical Development of Chaotic Communication Systems

Chua is most often credited with proposing the fundamental theory[2] and verifying the physical existence of chaotic attractors[26] in the early 1980s, which led to various circuit models representing the mathematical basis of chaotic circuits. An exceedingly simple circuit proposed by Matsumoto[27] implemented Chua's nonlinear resistor via a three-segment piecewise-linear resistor circuit that is pictured in Figure 5; this circuit produces a chaotic current/voltage relationship that is visible as the attractor shown on the right side of Figure 5.

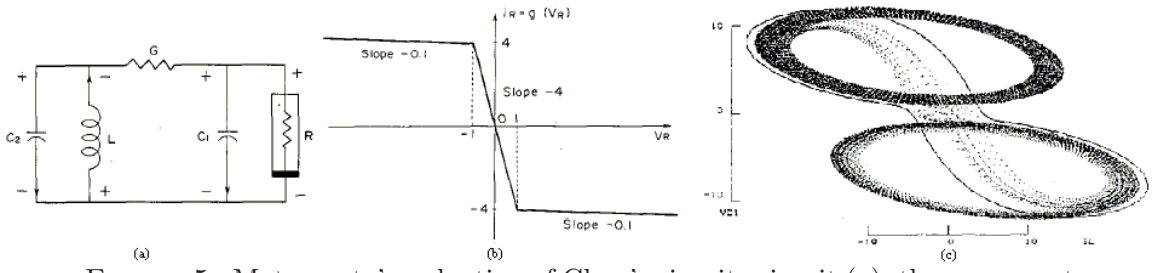

FIGURE 5. Matsumoto's reduction of Chua's circuit: circuit (a), three-segment piecewise-linear resistor (b), and chaotic attractor (c).

Various other chaotic circuits have been constructed, including monolithic passives[28], CMOS-based circuits[29], and improved active models[30]. The singular difficulty in constructing chaotic circuits is a method to synchronize two theoretically identical chaotic circuits such that they remain in the same state for an exploitable period of time. A 1990 paper by Ott, Grebogi, and Yorke[31] proposed a new method for controlling chaotic systems by

showing "one can convert a chaotic attractor to any one of a large number of possible attracting time-periodic motions by making only small time-dependent perturbations of an available system parameter." By being able to control chaos, engineering implementation of Shannon's noise-like sequences became realizable. Pecora and Carroll proposed a self-synchronous chaotic circuit[32] and a general methodology for synchronizing chaotic circuits in 1991[33]. The use of chaotic waveforms for robust communication systems soon gained popularity, with Wu/Chua proposing a synchronization method having application in secure communication systems[16]; Dedieu, Kennedy, and Hasler proposing a basis for chaos shift keying[3]; and Pecora and Carroll generalizing the synchronization methods for numerous applications in 1997[34]. The fundamental approach of the proposed systems is to construct a chaotic circuit that is then used to modulate a data signal, followed by a chaotic state synchronization scheme for a replica of the chaotic circuit at the receiver coupled with a demodulator. Some systems depended on a codebook of chaotic basis functions[35], others implemented a frequency diverse replica of the signal that could be used to re-synchronize the received waveform[36], and still others transmitted periodic chaotic state update information (or state-update algorithms)[37]. These systems have been demonstrated to satisfy the transmission characteristics of chaotic waveforms, yet lack the practical robustness or channel throughput of a maximal entropy waveform due to the chaotic state transfer overhead; further, various exploitation mechanisms have been identified that reduce the security of self-synchronous chaotic circuits.

The evolution of technology proposed for chaotic waveforms started with additive chaos masking[15] and chaotic shift keying[3] techniques in 1993. Additive chaos masking is a brute force approach to additively combining a low amplitude message signal (perhaps 20-30 dB down) with a chaotic signal; the performance of this system depends on the extremely low SNR of the message signal relative to the chaotic signal, limiting the amount of channel capacity utilized. Chaotic shift keying systems, also known as chaotic switching systems, use a data sequence to select from two or more chaotic circuits with similar chaotic attractors, but different state parameters; a correlation-based receiver uses a maximum likelihood estimate as to which symbol was transmitted each symbol period. From 1993 to 1995, the proposed chaotic communication systems generally used the chaotic sequence to modulate the data sequence consistent with chaotic synchronization schemes[38]. Chaotic parameter modulation[39] uses the data symbols to effect changes in the chaotic circuit via modulating the attractor used; the receiver attempts to recover the data sequence by tracking the changes in the chaotic attractor. In a slightly different version, chaotic non-autonomous modulation[16] directly perturbs a single chaotic attractor that evolves over time. More recently, chaotic communication approaches approximating more traditional digital modulation schemes have appeared[40]. Chaos phase

shift keying (CPSK) derived from the noncoherent format of selecting between chaotic basis functions to an evolving coherent chaotic topology, where both the transmitter and receiver maintain a chaotic circuit in the identical state; state update information is passed as part of the data stream and separated at the data link layer to maintain synchronization between chaotic circuit parameters. A differential chaos-shift keying (DCSK) approach uses a sequence of reference tones to aid correlation at the receiver; this reference tone inherently results in cyclostationary features that reduce the signal entropy and is susceptible to the channel effects of distinct frequency bands. Additional schemes including chaotic on-off keying, frequency modulated DCSK, correlation delay shift keying, symmetric chaos shift keying, and quadrature chaos shift keying have also been proposed.

In addition to the demonstrated limitations of specific chaotic communication systems, the complexity of the chaotic signal acquisition and chaotic circuit synchronization methods requires excessive amounts of the transmission bandwidth for chaotic circuit state update that practical communications is inhibited; in some cases[41], these synchronization algorithms asymptotically approach 90% of the transmission bandwidth. Nevertheless, active research has continued into the methodology of constructing chaotic circuits and using them to transmit information through a practical communications channel. Much of the analytical theory for BER curves, signal detectability, and predicted performance has matured since the early 1990s, but no practically implementable design has been published that effectively implements a chaotic communication system with comparable performance characteristics to traditional DS spreading systems (CDMA); the greatest challenge remaining is an efficient methodology for maintaining a synchronization lock between the chaotic circuits at the transmitter and receiver. In a 2003 publication[40] entitled "Chaos-Based Digital Communication Systems," Lao and Tse state

"chaos-based communication systems are still considered immature from the practical engineering standpoint... coherent detection requires the availability of synchronized replicas of the chaotic signals at the receiver. This in turn requires robust synchronization between the chaotic systems at the transmitter and the receiver. The problem is non-trivial, *and at the time of writing, there are still no acceptable chaos synchronization schemes that can be applied to communication systems for the required low signal-to-noise conditions.*" (p15)

Lao and Tse continue on to justify the increased interest in noncoherent chaotic communication systems since those are viewed as likely easier to implement. Their observation is

repeated in a 2007 publication[42] on chaos-based multiple access techniques:

"A number of coherent [chaotic communication] systems have been suggested and studied. For a coherent system, an exact replica of the chaotic signal needs to be reproduced at the receiving end. *Because robust synchronization techniques are not yet available, coherent systems are still not realizable in a practical environment.*" (p6)

The fundamental need for constructing a practically implementable coherent chaotic communication system is an acquisition and synchronization methodology that overcomes the practical aperiodicity of the chaotic evolution; we must learn to tame chaos.

1.2.3 Existing Models of Chaotic Circuit Synchronization

The nonlinear dynamics community has long realized the importance of a robust synchronization mechanism in constructing a practically implementable chaotic communications system. The generally accepted state of the art in chaotic circuit synchronization can be broken into three categories[42, 43], yet *none has been shown sufficient to produce a practically implementable coherent chaotic communication system*[40, 42, 44].

► **Identical synchronization**, also known as Pecora-Carroll synchronization, assumes a specially conditioned pair of chaotic circuits (strongly parameter matched) where the state of the receiver converges asymptotically to the state of the transmitter. More precisely, given a dynamical system at the transmitter with current state x, $\dot{x} = f(x)$, and a (nearly) identical replica of this dynamical system with current state x', $\dot{x}' = f(x')$, identical synchronization occurs when

$$\lim_{t\to\infty} ||x'(t) - x(t)|| = 0$$

for any combination of initial states $x(0)$ and $x'(0)$[43]. These synchronization mechanisms rely significantly on underlying circuit and transmission channel assumptions and fail to support robust synchronization as required for practical communications[43, 40, 42, 32].[3]

► **Generalized synchronization** is a weaker form than identical synchronization where instead of asymptotic convergence between the two chaos states, the receiver is asymptotically

[3]Among the assumptions in Pecora-Carroll synchronization, the two dynamical systems must have Lyapunov exponents that are negative for stability (considerably similar to eigenvalues), which implies a passive system, chaotic systems with identical parameters (circuit elements match), and that the initial conditions of the two systems are sufficiently close. The synchronization method is essentially an open-loop estimation technique, which cannot be robustly constructed in a noisy transmission channel.

convergent with a transformation of the transmitter state. More precisely, for a transformation M,

$$\lim_{t\to\infty} ||x'(t) - M(x(t))|| = 0$$

There is no requirement that the transformation M be invertible, so the determination of the transmitted chaotic basis function (i.e. the user data) is not guaranteed even if the two nonlinear dynamical systems exhibit generalized synchronization[43, 45].

▶ **Phase synchronization** is an even weaker form of convergence where the phases of the two chaotic states evolve to a bounded difference, where phase $\phi(t)$ is treated as a monotonically increasing function of time. More precisely,

$$\lim_{t\to\infty} ||\phi'(t) - \phi(t)|| \leq C$$

Highly stable chaotic systems like the spiral Chua attractor[46] show promise when combined with error-feedback synchronization controls. In these feedback mechanisms, the instantaneous difference between the received signal and the internal replica of the chaotic basis function is treated as an error term for insertion to a control loop; these methods are satisfactory in the analytical sense, yet have been shown to respond poorly to random perturbations (i.e. channel effects or AWGN) that are present in communication systems.

1.2.4 Chaotic Data Modulation

Various methods have been employed for encoding a data stream using a chaotic signal. The additive chaotic masking (additive combination of a low-level data signal and a chaotic sequence) is relatively ineffective in hiding the data signal statistical characteristics. Chaotic shift keying, or selecting between two distinct chaotic sequences based on the data symbol, is an effective method and may be combined with additional methods for incremental improvements; the downside of chaotic shift keying is that the receiver is necessarily noncoherent since it does not know which chaotic circuit is being employed. Chaotic parameter modulation, or adapting the characteristics of the chaotic generator based on the user data, is potentially risky if the noncoherent chaotic circuit evolution impacts synchronization. Chaotic parameter modulation schemes are equivalent to chaotic shift keying systems for constrained digital chaotic sequences, since a comparable noncoherent reception technique will be employed. Differential chaos shift keying transmits a duplicate of the chaotic signal to the receiver on either a different frequency channel or by using time-division protocols; this technique is clearly bandwidth inefficient since redundant chaotic state information must be sent independently to receive user data on the primary channel.

The three most promising variants of chaotic modulations[5, 44, 41] from a synchronization and noise performance perspective are chaotic phase shift keying (CPSK), which is comparable to direct sequence spreading[43], chaotic on-off keying (COOK), where data is transmitted similar to standard OOK with the presence or non-presence of a waveform within the symbol timeslot, and differential chaos shift keying (DCSK)[40], which embeds the evolving chaotic state characteristics into the waveform itself. Variants of CPSK modulations may be detected with both coherent and noncoherent receivers[43]. In the case of a coherent communication system, either a correlation receiver or a matched filter receiver (difficult to implement due to changing coefficients of the matched filter) employs a sufficient replica of the transmitter's chaotic basis functions to detect a transmitted symbol. Noncoherent receivers are easier to implement (the correlation is embedded into the receive signal), and provide better performance if synchronization cannot be maintained; this embedded correlation at the heart of noncoherent reception represents a fundamental deviation from the ideal maximal entropy waveform. Further, the noncoherent reception assumptions[43, 40] imply a positive spread SNR, which significantly limits a spread spectrum signal's potential applications. Another acceptable method for digital chaotic modulation is a chaotic frequency hopped local oscillator. Employing a chaotic sequence (or a continuous analog chaotic circuit) in the carrier frequency hop algorithm is little different than employing a good PRNG; during the pauses in frequency hops, the emitted waveform will retain all of its cyclo-stationary features. Therefore, the chaotic phase shift keying approach is seen as the best method for practically implementing a coherent chaotic waveform[40], with an overall modulation scheme that most nearly resembles arbitrary PSK symbols. Extending this chaotic phase shift keying to arbitrary signal constellations including amplitude modulation without inducing additional entropy reducing features is a non-trivial process, yet a novel solution is provided later in this dissertation.

A notional modulator block diagram and signal characteristics for a chaotic phase shift keying waveform are shown in Figure 6.

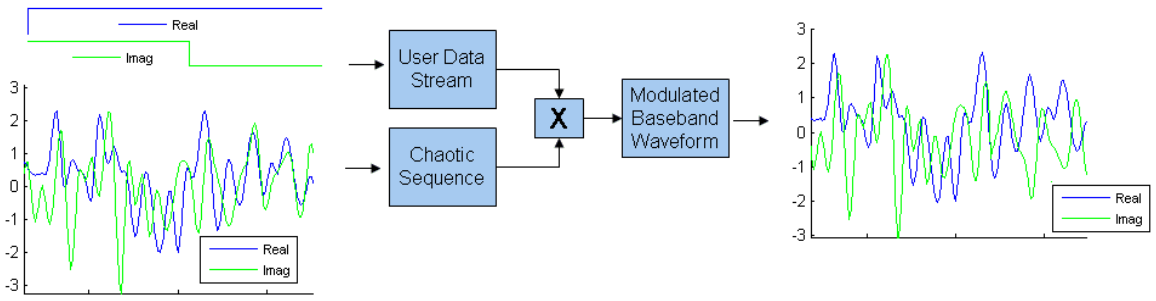

FIGURE 6. Signal characteristics for chaotic phase shift keying waveform.

Performance characterization of coherent chaotic communication systems by Kolumban and Kennedy[44] have demonstrated that "the noise performance of antipodal [chaos phase shift keying] can theoretically reach that of BPSK," which is "the best possible noise performance which can be achieved by any digital modulation scheme over an AWGN channel." However, "none of the chaotic synchronization techniques which exist in the literature is sufficiently robust to permit augmentation of the signal set in this way [for coherent chaotic communications]." Given that the performance characteristics of a coherent chaotic communication system using a correlation-based coherent receiver *could* provide the equivalent noise performance to traditional BPSK or QPSK variants, and yet retain the maximal entropy signal characteristics (impulsive autocorrelation and AWGN-like behavior) of a chaotic attractor, the development of a coherent receiver technology, including robust chaotic synchronization, remains an ideal waveform for use in a communication system. Fundamentally, the question is how do we tame chaotic circuits sufficiently well that they may be applied to this concept of a coherent communication system.

1.2.5 Measures of Signal Entropy

Various methods exist for qualifying and quantifying the entropy, or randomness, of an electrical signal. Secure communication systems that are often qualified by low probability of interception or low probability of detection, which indicates an absence of exploitable features from which to characterize the signal. By definition, a maximal entropy noise-like waveform that satisfies Shannon's criterion should have no detectable features, independent of signal-to-noise ratio. Stated simply, features represent reduced signal entropy. As a result, the quantifiable measures used for signal detection apply directly to quantifying the deviation of a signal from ideal AWGN, or decreased entropy.

Blind source separation (BSS) and related algorithms often use higher-order statistical methods (cumulants including skewness and kurtosis), matrix operators (discriminating signals via eigenvalue analysis), fast Fourier transforms, delay-and-multiply correlations, or autoregressive integral moving average (ARIMA) models. Each of these measures provide analytically equivalent methods to quantify the features or entropy of a waveform. As a simple example, consider the comparison of histograms for the first difference of a chaotic spreading sequence and a traditional pulse-shaped DS spreading sequence as shown in Figure 7.

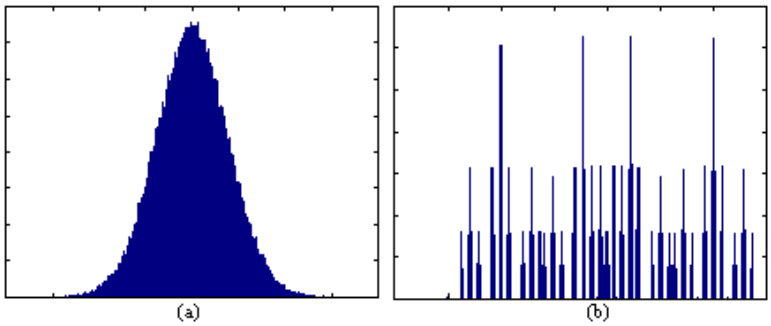

FIGURE 7. Histograms of first differences for chaotic (a) and traditional DS (b) spreading sequences.

Without too much surprise, the first difference of the chaotic spreading sequence has a Gaussian characteristic (the sum or difference of independent Gaussian random variables is Gaussian), which is no different than that expected of randomly sampled background noise. The transition regions of the traditional DS spreading sequence result in pronounced non-random statistical distributions, leading to an easily detectable sequence rate estimate, and therefore exploitable waveform features that signify non-maximal signal entropy. The quantitative characteristics of an ideal featureless waveform under the various measures are discussed in Chapter 2.4; in many cases, the measures represent equivalent mathematical processes using different fundamental frames of reference. The goal for a featureless chaotic waveform is a noise-like construction; any signal that is indistinguishable from bandlimited Gaussian noise would fit this specification.

▶ **Statistical chacteristics**: An ideal Gaussian noise-like sequence X with variance/power σ_X^2 should have cumulants of all orders that match that of AWGN. Further, all linear combinations of delayed samples (difference equations) should have a Gaussian probability distribution.

▶ **Matrix Operators**: An ideal signal will be resistant to direction finding, blind source separation, or other eigenvalue analyses that benefit from measurable signal features other than power detection or cross-correlation of spatially diverse receivers.

▶ **Spectral characteristics**: An ideal signal will have a perfectly flat frequency spectrum, with long-term DTFT computations yielding no significant spectral components. The waveform will also have a sharp bandpass response to prevent filter-induced correlations from aliasing, with all out-of-band spectral content maintained as white as possible.

▶ **Time-Domain Correlations**: An ideal signal will have an impulsive autocorrelation function (no distinguishable correlations with itself outside of a one chipping sequence period relative delay) and not fit within any statistically valid ARIMA models. The signal should

also have a negligible cross-correlation between in-phase and quadrature components to prevent I/Q imbalance in the RF upconversion stages from inducing correlations.

1.3 Overview of Dissertation

The goal of this dissertation is to document the conceptual development, design, simulation, and physical implementation of a practical coherent chaotic communication system. This system is characterized relative to the predicted performance that is in the open literature, contributing to the engineering techniques and chaotic communications architecture development via simulation models and hardware measurements. Extensions of this basic chaotic waveform are then introduced to document a broader class of maximal entropy noise-like waveforms, each with specific advantages.

▶ **Chapter 2** introduces examples of analog and digital chaotic circuits, characterization methods and a comparison of each, and ultimately the motivation for the use of digitally generated chaotic sequences for chaotic communications. This chapter outlines a novel approach for generating digital chaotic sequences based on engineering approximations to number theoretic techniques. The basic method employs the Chinese Remainder Theorem to combine sequences generated using irreducible chaotic polynomials calculated over distinct Galois Fields of mutually prime characteristic. Extensions to this method are provided to efficiently implement sequence lengths on the order of googols. Finally, an efficient implementation of the Box-Muller transformation is introduced for re-shaping the digital chaotic sequence as quadrature Gaussian noise for use in chaotic communications.

▶ **Chapter 3** provides a system level overview of the design, simulation, implementation, and measured hardware results of what is believed to be the world's first practically implementable coherent chaotic communication system. Much of this detailed implementation was performed in a collaborative environment by the Harris IR&D team, and is presented primarily as evidence of the practical applicability of the digital chaotic sequences in communication systems. Specific implementation details and models are excluded as are higher level protocol functionality to protect proprietary interests.

▶ **Chapter 4** provides a detailed analysis of distributed digital chaotic circuit synchronization, modifications to the traditional direct sequence spread spectrum approaches for chaotic waveform acquisition and tracking, a nonlinear selective noise cancellation approach and energy shaping mechanisms for heteroskedastic waveforms, a novel adaptive correlator design/implementation, techniques for improving transmission security, and improved efficiency signal processing techniques for digital chaotic waveform generation/reception.

▶ **Chapter 5** explores the theoretical, simulated, and measured coherent chaotic communication performance in live transmission mediums. In particular, performance in flat fading channels, multipath conditions, channel equalization, and RAKE reception are discussed. Communications based on noise-like chaotic waveforms are shown to exhibit superior noise and channel distortion performance when compared to direct sequence spread spectrum systems.

▶ **Chapter 6** describes a generalization of the basic noise-like chaotic waveform that provides options for lower transmitted peak-to-average-power ratios (PAPR). A basic technique for modulating the PAPR between extreme cases of the constant amplitude zero autocorrelation (CAZAC) waveform (approx 3 dB) and the standard chaotic waveform (approx 13 dB) show potential uses in environmentally adaptive cognitive radios, precision pseudo-ranging, and application aboard power-constrained platforms.

▶ **Chapter 7** introduces a conceptually new waveform dubbed "chaotic QAM" that retains all characteristics of the standard chaotic waveform, including Gaussian characteristic, yet permits adaptation to traditional amplitude-based modulation formats. Applicability of this chaotic QAM is discussed for increased channel throughputs, secure transmission, and extension to arbitrary chaotically spread signal constellations.

▶ **Chapter 8** presents a system-wide outline for chaos-based multiple access communications, harnessing the various chaotic waveforms for use in multiple-user and permission-based communication systems. Evaluation of the channel re-use characteristics, the traditional CDMA power problem, and multipath performance prove digital chaotic communications to be a viable, and in many cases technically superior, option for mass communications.

▶ **Chapter 9** closes out the dissertation by summarizing areas on ongoing research and presenting possible problems for future research. In particular, a simple technique for environmentally adaptive chaotic communications is presented in addition to generalizing the orthogonal chaotic signaling basis.

▶ **Appendix A** contains various Matlab scripts and support documentation for the analysis and simulation results presented in this document.

▶ **Appendix B** contains a listing of pending U.S. patents related to the work presented in this dissertation.

Specific contributions of the present work include:

Discrete-Time Discrete-Amplitude Chaotic Sequences: The fundamental difference between the system proposed in this dissertation and most in the literature is a fully discrete-time discrete-amplitude chaotic circuit constructed using chaotic polynomial evaluations within finite algebraic structures. The chaotic polynomials form ring generators that are

combined via residue number system (RNS) arithmetic and efficiently implemented in DSP primitives. These chaotic sequence generators then provide a means for constructing the prototype communication system. Various techniques for expanding the finite sequence repetition period and increasing the security of the chaotic sequence from reverse engineering are also provided.

Adaptive Correlation Techniques: An adaptable correlator topology is developed to provide rapid signal acquisition and robust signal tracking characteristics. This correlator serves for both the chaotic signal acquisition mode and demodulation mode; in the acquisition mode, adjustable thresholds provide Bayesian estimates of an intermediate correlation to reduce overall correlation processing.

End-to-end Simulation Results for Coherent Chaotic Communications: The entire design for a single-channel coherent chaotic communications system was implemented in Simulink and Synplify DSP in order to evaluate performance under various transmission conditions. Discussion is provided for a model-based synthesis approach, which has proven to be an efficient method for transferring the concept to hardware.

Hardware Implementation and Measurements: This dissertation discusses the design, implementation, and hardware measurements of what is believed to be the first functional hardware prototype of a practical coherent chaotic communication system. Significant effort has been invested in ensuring the developed structures are hardware efficient, while providing tools for characterizing the true performance of a physical system in addition to validating the simulation models. The final model is a TRL-6 prototype that can be leveraged for critical design of numerous chaos-based communication systems[47].

Modifications to Traditional Spread-Spectrum Receiver Processing: Numerous techniques and hardware implementations are provided to make the distinctions between structures used for processing traditional direct sequence spread-spectrum signals and those required for a chaotic signal. Phase, frequency, and time (sequence) tracking loops are implemented and discussed in detail, validating the use of early-late detection and traditional phase tracking techniques in coherent chaotic communication receivers.

Low-Level Protocol Security Enhancements: Various techniques have been developed to optimize the signal processing of the chaotic waveform, including corrections for the non-stationary modulated symbol energy, SNR improvements based on a priori knowledge of the coherent chaotic sequence, modulation techniques for optimized BER in dynamic channels, and methods for ensuring user-based security in multiple access systems.

Chaotic Waveform Variants: A fundamental difference between analog and digital communication systems is the ability to increase bandwidth efficiency via higher capacity

modulation schemes like QAM or 16APSK; these techniques are adjusted to create featureless coherent chaotic modulation of any arbitrary data constellation. Additional trades providing options for environmental adaptation of the chaotic waveform in the presence of interferers are also presented as a practical CAZAC waveform. Finally, the impulsive autocorrelation characteristic of the chaotic waveform is harnessed to construct a practical RAKE receiver that mitigates multipath images in the transmission channel.

Chapter 2: Generation of Chaotic Sequences

The core of any chaotic communication system is the pair of synchronized chaotic circuits at the transmitter and receiver. Whether an analog derivation similar to Chua's models, or a fully discrete-time discrete-amplitude version as discussed in this dissertation, the two chaotic circuits must remain sufficiently synchronized that the chaotic modulation process converting a data symbol into a chaotic signal can be inverted at the receiver to recover the data. Most attempts at constructing chaotic communication systems rely on analog circuits, building on Chua's original work[2] that showed electrical circuits can implement chaotic ordinary differential equations. Chua's circuit used a piecewise linear resistor as the key circuit element to implement chaotic properties; researchers have since employed different fundamental circuit elements to create chaotic behavior with various advantages and disadvantages. The basic difficulty of synchronizing "identical" copies of the same circuit in an efficient manner have limited the practical applicability of these analog chaotic circuits for communications. Additional research into engineering approximations of chaotic circuits, such as digitally implemented tent maps and chaotic polynomials have shown promise for their use in communications, yet with limitations that come from finite arithmetic. This chapter explores various approaches to constructing chaotic circuits and discusses the applicability of the circuits to communication systems. The chapter begins with a summary of the traditional approaches to constructing analog chaotic circuits, advances to performance of digital chaotic circuits, and concludes with a detailed construction of a novel digital chaotic circuit that has been emperically proven successful as a robust solution for coherent chaotic communication systems.

2.1 Analog Chaotic Circuits

The construction of analog chaotic circuits began in 1980 with Chua's detailed evaluation of nonlinear RLC networks[2] and their ability to dynamic nonlinear networks. The analytical condition for *strongly locally passive* resistors in these RLC networks led to Chua's 1984 construction of the first recognized chaotic circuit[26] that used a strictly passive piecewise linear negative resistor for the nonlinear circuit element. The same year, co-researcher Matsumoto proposed a simplification of Chua's original circuit[27] that implemented a different piecewise linear resistor. These circuit were ensured to be chaotic based on the properties of Poincaire return maps[48] by implementing the coupled differential equations:

$$\dot{x} = \alpha(y - \phi(x)) \qquad \dot{y} = x - y + z \qquad \dot{z} = -\beta y \qquad\qquad \alpha, \beta \in \mathbb{R}, \quad \phi(x) = m_1 x + \frac{m_0 - m_1}{2}(|x+1| - |x-1|)$$

An important observation from these original circuits was that although the existance of chaotic attractors was proven, the specific attractors exhibited by Chua and Matsumoto's circuits were different from each other and also previous work by Lorenz[49] and Rossler[50], leading to the belief that a wide class of chaotic system could be constructed with analog circuits. More importantly, the construction of circuits that exhibit chaotic behavior may be explored with any combination of nonlinear circuit elements that implement systems of chaotic differential equations. Over the next two decades, many creative approaches for implementing analog chaotic circuits, including implementation of the nonlinear elements by diodes and opamps[51, 52], ferroelectric coupling[53], optical circuits[54], voltage and current mode feedback[55, 56], and mutually cross-coupled capacitors[57, 58]. Significant limitations to these approaches have been identified, with the susceptibility to noise[59] and other electromagnetic interference effects[60] by the Chua circuit inhibiting the repeatability of the chaotic circuit evolution. That repeatability is required not only of the same physical circuit, but of two "identical" circuits that account for the practical device tolerances, before practical communication systems may be constructed with the chaotic circuit as an underlying process.

Synchronization of distinct chaotic circuits became the next goal, with Pecora and Carroll defining basic approaches in 1990[32]; Ott and Grebogi outlined initial approaches for synchronizing chaotic communications[31] the same year. The process of accurately setting and synchronizing the chaotic state of two independent chaotic circuits coherently was immediately recognized as a barrier to chaotic communications. Various approximations to the construction of chaotic circuits emerged, leading to discrete-time nonlinear circuits[61] and implementations specialized to evaluation of higher-order attractors[62] or Lyupanov exponents[63]. Nevertheless, despite the widespread recognition that chaotic signals have superior performance potential[64, 65, 66, 67, 68, 14, 3, 69], the prevailing consensus has been that "because robust synchronization techniques are not yet available, coherent [chaotic] systems are still not realizable in a pratical environment[42]."

2.2 Discrete Chaotic Circuits

The undeniable potential of using chaotic circuits in practical communication systems has expanded the search to both discrete-time and discrete-amplitude chaotic circuits that possess some if not all of the chaotic properties of analog chaotic circuits. Discrete-time circuits rely on the mathematical analogy between continuous differential equations and finite difference equations; this conversion is not generally allowable, yet specific engineering approximations to ODEs have been proven to exhibit chaotic properties. One common example is the chaotic

tent map whereby successive values are calculated via the piecewise linear mapping:

$$x_{n+1} = \begin{cases} \mu x_n & x_n < \frac{1}{2} \\ \mu(1 - x_n) & x_n \geq \frac{1}{2} \end{cases} \qquad \mu \in \mathbb{R}$$

To present a chaotic mapping, the choice of μ must be such that the domain and range of the mapping are dense (e.g. $\mu = 2$ on $[0, 1]$). More complex mappings, such as the logistic map first defined by Verhulst[24] as

$$x_{n+1} = r x_n (1 - x_n)$$

can also be proven as chaotic for proper choices of r. These discrete time mappings are useful for analysis and quick generation of "chaotic" sequences, but are limited in their susceptibility to roundoff errors in practical computational environments. Effective implementations for chaotic pseudorandom number generation[70, 71] have been identified, but both are dependent on the hardware platform or rounding rules, operating on a binary subset of the target domain. The search for discrete-time chaotic systems, and more specifically, their applicability to communication systems has continued, with work by Andreyev and Efremova proving the ideal signal separation capabilities of chaotic signals[72]; it is believed that the choice of the relatively simplistic logistic map for their separation algorithm leads to higher than necessary SNR requirements. Better mappings, such as chaotic polynomial computations, eliminate much of the internal correlations of the signal and are more useful in wideband communication systems; the most common instantiations of these polynomial maps are in specialized filter structures[73, 74] and, more recently, finite algebraic structures[75, 76]. The latter offers a unique opportunity by eliminating the roundoff error problem inherent to most discrete-time chaotic circuits. The only limitation is that quantification of traditional chaotic parameters like attractors or Lyupanov exponents lose meaning; from an engineering perspective however, we can loosely build the bridge between digital chaotic circuits and their applicability to communication system by proving they exhibit maximal entropy, and thus satisfy Shannon's criterion[1] for noise-like waveforms that fully utilize the channel capacity.

The practical goals in creating a digital chaotic circuit are:

1. Chaotic behavior indistinguishable from mathematically precise nonlinear dynamics
2. Efficient hardware implementation and control mechanisms
3. Efficient chaotic state initialization and synchronization
4. Practically infinite number of chaotic states

The compromise between these objectives is a fully digital solution with dynamic range exceeding that of the transmitter's analog components (the digital-to-analog converter is a

logical bottleneck at 60-80 dB of dynamic range) and a signal bandwidth that can be efficiently interpolated and processed at baseband (limited primarily by digital logic design, filtering rates, and data converters to perhaps 50-100 MHz of signal bandwidth). Provided both of these conditions may be met within the measurability of a practical receiver, then the chaotic waveform is chaotic enough.

The dynamic range is solved by constructing the digital chaotic sequence generator with a repetition period of much greater than $\log_2 10^{\frac{(80\text{ dB})}{10}} \approx 27$ bits; to compare simulation and hardware results precisely with standard "double" formats, the state of a digital generator must be updated periodically to account for and correct roundoff errors in the mantissa. Moreover, if the repetition period is made longer than a scaled inverse of the fundamental charge on an electron[20] ($\frac{a}{1.6 \cdot 10^{-19}} \approx 2^{62.4}$),[4] then the digital system should be approximately equivalent in 'chaotic-ness' as any analog circuit after digitization. The search for such a digital chaotic circuit with the appropriate chaotic properties and applicability to a chaotic communication system, marks the beginning of the doctoral research presented in this dissertation.

2.3 A Novel Digital Chaotic Circuit

The preferred approach to constructing a practically implementable digital chaotic circuit builds on the recognition that finite algebraic structure can both effectively implement a chaotic mapping and elude roundoff errors by use of closed operators[75, 76]. These finite algebraic structures are not typically efficient in digital hardware unless processed over rings or fields of binary characteristic; generalizing the approach though to p-adic spaces opens a wide range of number theoretic techniques. Since chaotic systems are significantly impacted by small errors, synchronization of a weighted number system approach requires constant attention and correction for the rounding errors that occur during any calculation. An alternative is the residue number system (RNS)[5] that belongs to classes of finite algebraic structures (e.g. rings, Galois fields) and only implement closed operators on those finite structures; RNS based arithmetic will always return an error-free result without the need for error accumulators or periodic re-synching of the sequence index.

To explore the development of this novel digital chaotic circuit in more detail, the first subsection begins with a tailored overview of the relevant definitions and algebraic/number theoretic vocabulary followed by construction of a distributed chaotic polynomial. Algorithmic

[4] As an extreme example, common electrolytic capacitors begin to break down when junction voltages reach the ballpark of 100V; for Chua's circuit with an excessively large capacitance of 2 Farads[27], the implied charge storage capacity is $Q = CV = 200$ Coulombs, or $1.25 \cdot 10^{21} \approx 2^{70}$ electrons.

[5] A residue number system may be viewed most simply as an n-tuple of independent digits that each operate with their own specific characteristics, while a weighted number system is inherently linked by the rollover over one digit cascading into other digits.

optimization of this chaotic polynomial is then discussed with concern both for the hardware utilization and chaotic state parameterization. The first subsection concludes with detailed description of techniques to increase the digital chaotic sequence length. The second subsection applies the developed tools to create a prototype digital chaotic sequence generator, including a generalized non-chaotic masking sequence that prevents reverse engineering. The final subsection in this chapter provides a detailed evaluation of the prototype chaotic sequence generator under various feature detection measures to determine whether the sequence does in fact meet the maximal entropy criterion for use in a chaotic communication system.

2.3.1 Residue Number Systems

Residue number systems (RNS) have been studied for over 2000 years as an intriguing alternative to weighted number systems. RNS generalizes the traditional decimal number system that employs an n-tuple of digits with associated weights 10^k by allowing mixed-radix number representations. In particular, RNS constructions using $GF(2^k)$ have been studied for arithmetic in digital hardware[77, 78], while specific applications of mixed-radix systems have been studied for Fourier transforms[79, 80], RNS polyphase filtering[81], and even quantum circuits[82]. General references for number theory[83, 84] and its application in signal processing[85, 86] are widely available; the research in RNS for communication circuits yields benefits in certain applications, while general use is hampered by the computational effort to convert to and from a weighted number system.

2.3.1.1 Algebraic Rings and Fields

A ring[87] is a set S with two binary operators[6] $+$ and $\cdot$ satisfying the six conditions (1)-(6).

1. **Additive associativity**: $\forall a, b, c \in S, \quad (a + b) + c = a + (b + c)$
2. **Additive commutativity**: $\forall a, b \in S, \quad a + b = b + a$
3. **Additive identity**: $\exists\, 0 \in S \; s.t. \; \forall a \in S, \quad 0 + a = a$
4. **Additive inverse**: $\forall a \in S, \quad \exists (-a) \in S \; s.t. \; a + (-a) = (-a) + a = 0$
5. **Left and right distributivity**: $\forall a, b, c \in S, \quad a \cdot (b+c) = a \cdot b + a \cdot c$ and $(a+b) \cdot c = a \cdot c + b \cdot c$
6. **Multiplicative associativity**: $\forall a, b, c \in S, \quad a \cdot (b \cdot c) = (a \cdot b) \cdot c$

A finite algebraic structure that satisfies the additional three conditions (7)-(9) is termed a field; if the field contains finitely many elements, then it is termed a finite field or Galois

[6] A binary operator is defined on a set S and takes two elements from S as inputs, returning a single element of S.

field (GF)[88].

7. **Multiplicative commutativity**: $\forall a, b \in S, \quad a \cdot b = b \cdot a$
8. **Multiplicative identity**: $\exists\, 1 \in S \; s.t. \; \forall a \in S, \quad 1 \cdot a = a \cdot 1 = a$
9. **Multiplicative inverse**: $\forall a \neq 0 \in S, \quad \exists a^{-1} \in S \; s.t. \; a \cdot a^{-1} = a^{-1} \cdot a = 1$

Throughout history, cryptographic techniques have relied on the properties of rings and Galois fields with prime characteristics (sizes) to enact the properties of number theory. These properties will be referenced loosely as ring/field properties and implemented as various discrete mappings – oftentimes within an isomorphism of the precise mathematical definitions to retain the proper statistical properties and to simplify the hardware implementation. The notation used for a finite field (or Galois field) of characteristic p in this dissertation is $GF(p)$ to denote the set $\{0, 1, 2, 3, \ldots, (p-1)\}$ with the associated properties 1-9 when all arithmetic is processed modulo p.

The primary significance of the finite algebraic structures for chaotic communications is that they provide an efficient manner to create extremely long pseudo-random number generators (PRNG) that can in turn be used to mimic a chaotic sequence. As an example, consider the associated behavior of a mapping on a ring of characteristic 2^k [7] for satisfying the definitions of a chaotic system.

► **Denseness:** For the discrete mapping to most closely approximate the continuous idea of "dense," it must take all possible values within the set; i.e. the mapping must be exhaustive. Further, the mapping must be able to result in a value that is *arbitrarily* close to an chosen point in the set; since the finite domain/range are identical and discrete in nature, the chosen element must be within the set.[8]

► **Sensitivity to initial conditions:** The chosen mapping on a finite algebraic structure should be nonlinear enough that small perturbations in the input result in significantly different output values over time. Since the discrete algebraic structure is finite, any change in input value will also be discrete; the effective change of a single perturbation will be an advancement of the mapping to somewhere in the next periodic cycle, resulting in a practical desire to make the sequence repetition period (or domain/range sizes) as large as possible.

[7] Strictly speaking, product fields do not have multiplicative inverses and therefore cannot be "fields;" the interpretation of a finite-precision binary number system will shift between $GF^k(2)$ and $\mathbb{Z}_{2^k}$, which can trivially be connected via a bijective mapping.

[8] A better approximation for this is a progression of Cantor-like subsets[18, 19] S_n on the domain [0,1]. Defining $S_1 = \{0, \frac{1}{2}\}$, $S_2 = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}\}$, $S_3 = \{0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}\}$, etc., we can imagine a better approximation to *arbitrary* closeness by simply increasing the value k, or equivalently, the number of bits in the binary representation.

► **Topological transitivity:** In simplest terms, the concept of topological transitivity ensures that inputs which are arbitrarily close can evolve into significantly different mappings. Any mapping rule on a finite algebraic structure that ensures the ability to get from any point in the domain to any other point in the domain satisfies topological transitivity in spirit if not mathematical precision.

A simple example that meets all of these conditions for a digital chaotic circuit is the mapping $f(x) = 3x^3 + 3x^2 + x + 7$ on the field $GF(11)$. Another polynomial exhibiting chaotic behavior is $T_3(x) = 4x^3 - 3x$[23], yet this mapping fails practically due to a fixed point at $x = 0$. These discrete chaotic systems may be constructed as semi-autonomous *ring generators* that compute evaluations of an irreducible chaotic polynomial $f(x)$ within a prime residue space of characteristic p_i; a notional block diagram of a ring generator is shown in Figure 8.

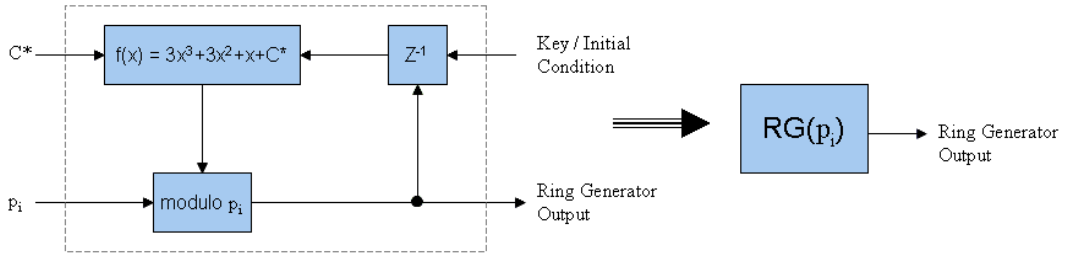

FIGURE 8. Notional diagram of irreducible polynomial computation in a ring generator.

The overall purpose of these operations and ring/field criteria is to demonstrate that under specific conditions, a suitable approximation to chaotic system behavior may be obtained by completely finite structures, reducing further need to distinguish between the physics of a discretized world and the mathematics of continuous chaotic systems.

2.3.1.2 Chinese Remainder Theorem

Another attraction to operating within a residue number system as opposed to a weighted number system is the ability to efficiently implement hardware operations independently within each residue space and then combine via the Chinese Remainder Theorem (CRT). The CRT was first published in Sunzi's *Book of Arithmetics* nearly two millennia ago as depicted in Figure 9.[9]

[9]The answer is 23 (=≡).

今有物、不知其數。三、三數之，賸二；五、五數之，賸三；七、七數之，賸二。問物幾何?
答曰：二十三。
術曰：〝三、三數之，賸二〞，置一百四十；〝五、五數之，賸三〞,置六十三；〝七、七數之，賸二〞，置三十。并之，得二百三十三。以二百一十減之，即得。
凡三、三數之，賸一，則置七十。凡五、五數之，賸一，則置二十一。凡七、七數之，賸一，則置十五。一百六以上，以一百五減之，即得。

FIGURE 9. Original statement of the Chinese Remainder Theorem by Sunzi.

Sun Zi's method was generalized by Qin Jiushao in his 1247 *Mathematical Treatise in Nine Sections*[83] to give a method for combining elements of distinct (mutually prime) residue spaces into an equivalent element modulo the product of the residue space characteristics. More precisely, consider any n distinct prime numbers, $p_1, p_2, ..., p_n$, whose product is $M = \prod_{i=1}^{n} p_i$. For any element $x \in GF(M)$, the combined n-tuple results of $x \mod p_i$ for all possible primes will be unique for each x. The exact formula can be easily constructed, by writing the element x with remainders $r_1, r_2, ..., r_n$ when computed modulo primes $p_1, p_2, ..., p_n$. Note that the additional inverse terms are computed with respect to the p_i characteristics and not the overall combined characteristic, M; trivially, the ratio $\frac{M}{p_i}$ will have a nonzero remainder modulo p_i, ensuring a unique multiplicative inverse.[10]

$$x = \left(\frac{M}{p_1}\right) \cdot \left[\left(\frac{M}{p_1}\right)^{-1}_{(p_1)} r_1\right]_{(p_1)} + \left(\frac{M}{p_2}\right) \cdot \left[\left(\frac{M}{p_2}\right)^{-1}_{(p_2)} r_2\right]_{(p_2)} + ... + \left(\frac{M}{p_n}\right) \cdot \left[\left(\frac{M}{p_n}\right)^{-1}_{(p_n)} r_n\right]_{(p_n)} \mod M$$

Remembering that M is the product of all the primes, we notice that we may move easily from the element x to its remainders r_i. To reduce x to r_i, we just compute $x \mod p_i$.

$$\begin{aligned} x \mod p_i &= \sum_{j \neq i} \left(\frac{M}{p_j}\right) \cdot \left[\left(\frac{M}{p_j}\right)^{-1}_{(p_j)} r_j\right]_{(p_j)} + \left(\frac{M}{p_i}\right) \cdot \left[\left(\frac{M}{p_i}\right)^{-1}_{(p_i)} r_i\right]_{(p_i)} \mod p_i \\ &= p_i \cdot \sum_{j \neq i} \prod_{t \neq i,j} p_t \cdot \left[\left(\frac{M}{p_j}\right)^{-1}_{(p_j)} r_j\right]_{(p_j)} + \left[\left(\frac{M}{p_i}\right) \cdot \left(\frac{M}{p_i}\right)^{-1}_{(p_i)}\right]_{(p_i)} r_i \mod p_i \\ &= r_i \mod p_i \end{aligned}$$

In effect, the dynamic range increases from that of a single prime characteristic to that of the product of all such primes. This increase may be taken with arbitrarily many inputs (provided all are mutually prime), making CRT an effective and efficient method of expanding the finite size. The difficulty of using CRT in practice is computing all possible multiplications and inverse elements in real time: computations over a general prime number base in digital hardware are not typically an efficient process. In summary, the CRT is a strong analytical method to increase the effective dynamic range of a ring generator, efficiently permitting

[10]The notation $\left(\frac{M}{p_i}\right)^{-1}_{p_i}$ refers to the multiplicative inverse of the nonzero integer $\frac{M}{p_i}$ modulo p_i. Since $M = \prod_{j \neq i} p_j$ and the p_j terms are mutually prime, the result will always be nonzero.

mixed-radix computation, but the sequence remains highly deterministic if the prime numbers are known a priori (via extended Euclidean algorithm)[84].

2.3.1.3 Application of CRT Combination to a Chaotic Polynomial

The instinctive goal of the Chinese Remainder Theorem is to be able to do arithmetic in relatively small, mutually prime, residue spaces (p_i for $i = \{1, 2, \ldots, n\}$) and then combine to form the equivalent operation if done purely within the large product space ($M = \prod p_i$). A simple example is the calculation of the chaotic polynomial

$$f^*_{M=3,563,762,191,059,523}(x) = 3x^3 + 3x^2 + x + 97,903,550,178,815$$

More efficiently than trying to calculate a polynomial for any $x \in \mathbb{Z}_M$,[11] we can break up M into its mutually prime factors[12]

$$M = 3,563,762,191,059,523 = 251 \cdot 257 \cdot 467 \cdot 479 \cdot 491 \cdot 503 = p_1 p_2 p_3 p_4 p_5 p_6$$

If M is divisible by any squares (i.e. 2^2, 3^2, etc), then additional care must be implemented when calculating residues and recombining. Choosing p_i as distinct prime numbers (and therefore automatically mutually prime) to avoid this unnecessary complexity, the entire process may be reduced to the following set of coupled equations:

$$\begin{aligned} f_{p_1=251}(x) &= 3x^3 + 3x^2 + x + 39 & \qquad f_{p_2=257}(x) &= 3x^3 + 3x^2 + x + 110 \\ f_{p_3=467}(x) &= 3x^3 + 3x^2 + x + 15 & \qquad f_{p_4=479}(x) &= 3x^3 + 3x^2 + x + 233 \\ f_{p_5=491}(x) &= 3x^3 + 3x^2 + x + 202 & \qquad f_{p_6=503}(x) &= 3x^3 + 3x^2 + x + 8 \end{aligned}$$

where the constants are obtained as the appropriate modular reductions of 97,903,550,178,815 modulo $\{p_1, p_2, p_3, p_4, p_5, p_6\} = \{251, 257, 467, 479, 491, 503\}$.

$$\begin{aligned} 97,903,550,178,815 &\equiv 39 \mod p_1 & \qquad 97,903,550,178,815 &\equiv 110 \mod p_2 \\ 97,903,550,178,815 &\equiv 15 \mod p_3 & \qquad 97,903,550,178,815 &\equiv 233 \mod p_4 \\ 97,903,550,178,815 &\equiv 202 \mod p_5 & \qquad 97,903,550,178,815 &\equiv 8 \mod p_6 \end{aligned}$$

For an evaluation of $f_M(x = 123,456)$, the equivalent computation in the residue spaces is

[11] Note that the chosen M for this example is nearly as large as a standard mathematical analysis tool can represent accurately, based on the IEEE standard[89] for *double* data precision, having a 52-bit mantissa.

$$\lceil \log_2(M) \rceil = \lceil \log_2(3,563,762,191,059,523) \rceil = \lceil 51.662 \rceil = 52 \,\text{bits}$$

[12] Again, a note on efficiency is that there are no general methods for factoring a large number; computers can easily perform brute force searches for numbers in the neighborhood of 40-50 bits absent knowledge of specific properties of the number[90].

$$f_{p_1}(123,456 \mod p_1) = f_{p_1}(215) = 216 \qquad f_{p_2}(123,456 \mod p_2) = f_{p_1}(96) = 10$$
$$f_{p_3}(123,456 \mod p_3) = f_{p_1}(168) = 404 \qquad f_{p_4}(123,456 \mod p_4) = f_{p_1}(353) = 98$$
$$f_{p_5}(123,456 \mod p_5) = f_{p_1}(215) = 271 \qquad f_{p_6}(123,456 \mod p_6) = f_{p_1}(221) = 331$$

Applying the Chinese Remainder theorem, the multiplicative inverse terms are

$$\left(\frac{M}{p_1}\right)^{-1}_{(p_1)} = 150 \quad \left(\frac{M}{p_2}\right)^{-1}_{(p_2)} = 118 \quad \left(\frac{M}{p_3}\right)^{-1}_{(p_3)} = 97 \quad \left(\frac{M}{p_4}\right)^{-1}_{(p_4)} = 439 \quad \left(\frac{M}{p_5}\right)^{-1}_{(p_5)} = 81 \quad \left(\frac{M}{p_6}\right)^{-1}_{(p_6)} = 329$$

giving an overall output value of

$$\begin{aligned} f_M(x = 123,456) \equiv\ & (216 \cdot 150)_{p_1} \prod_{i \in \{2,3,4,5,6\}} p_i + (10 \cdot 118)_{p_2} \prod_{i \in \{1,3,4,5,6\}} p_i + (404 \cdot 97)_{p_3} \prod_{i \in \{1,2,4,5,6\}} p_i \\ & + (98 \cdot 439)_{p_4} \prod_{i \in \{1,2,3,5,6\}} p_i + (271 \cdot 81)_{p_5} \prod_{i \in \{1,2,3,4,6\}} p_i + (331 \cdot 329)_{p_6} \prod_{i \in \{1,2,3,4,5\}} p_i \\ \equiv\ & 2,179,107,969,003,004 \mod M \end{aligned}$$

A more general method for computing a chaotic polynomial in residue number spaces is depicted in Figure 10. Note that the ring generators $RG(p_1), \ldots, RG(p_n)$ are assumed to implement an appropriate reduction of $f_M(x)$ over distinct prime residues $p_1, \ldots, p_n$. The arithmetic for combining ring generator outputs can be viewed as occuring in Galois fields, resulting in an element of $GF(M)$, or equivalently an overall output modulo M.

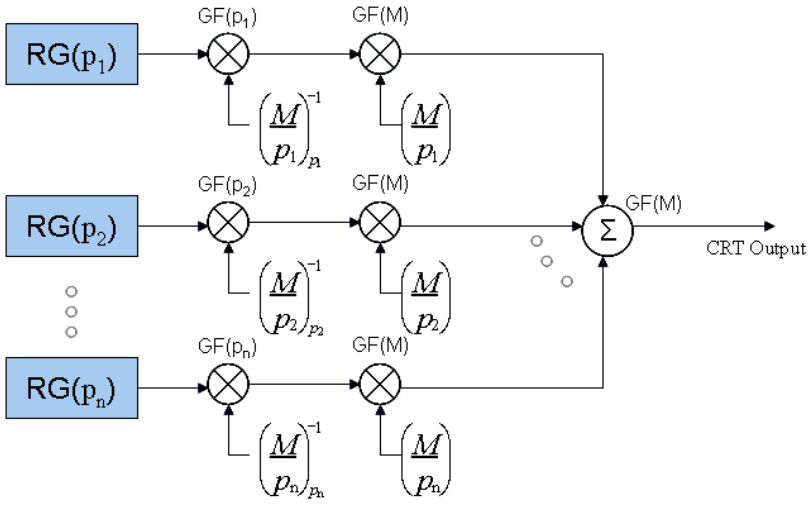

FIGURE 10. CRT implementation of chaotic polynomial computation.

2.3.1.4 Hardware Efficient CRT Substitution

Since digital logic operates with two fundamental logic states, the industry has invested significant effort in making binary arithmetic as efficient as possible; the nature of the arithmetic assists significantly since weighted binary number systems have so few states that they

may be hard coded in digital logic. Extending these efficiency gains to non-binary number systems (residue or weighted) is a traditionally difficult problem; specific algorithms like FFTs[79] and multiplication[91, 92] benefit from thinking in terms of non-binary number systems, yet these methods do not generalize for all applications. More importantly, arithmetic operations over large prime numbers are sufficiently difficult that it can provide a degree of security when only partial information is known[90].

For the preceding example of a chaotic polynomial computation over a large dynamic range, performing the modular reductions before and CRT after each calculation is extraordinarily cumbersome. For example, the initial multiplications at the output of the ring generators each require a unique modular reduction before passing to the next set of multipliers. These multipliers will not require a modular reduction (the product will never exceed M), but will require multiplication (shift-adds) with one term equal to the full dynamic range of M. The overall output addition includes a final modular reduction modulo M, which requires extremely wide adders.

Instead, we note that the chaotic polynomial must be topologically transitive (approximated by an exhaustive mapping, or equivalently, choice of an irreducible chaotic polynomial in each residue space) and only perform the modular reductions when initializing the system. Allowing each ring generator to evolve separately, the fidelity of the subsequent CRT computation may be compromised since the chaotic polynomial computations do not depend on feedback of the CRT output; this reduction should retain the desired statistical chacteristics of the portion of chaotic polynomial, yet has no need to retain unused precision. As an example, if the chaotic polynomial provided in the previous section were computed accurately each cycle, the result would be a binary representation with 52 bits precision. Use of these values as a chaotic spreading sequence in a chaotic communication system will be ineffective beyond the range of the digital-to-analog converter, making a truncated selection of 16 bits sufficient. Further, knowledge that all subsequent computation will be more efficiently performed in traditional binary number systems, this enforces an effective modulo 2^{16} operator by discarding all but the bottom 16 bits in intermediate calculations (propagating the modulo 2^{16} operation backwards).

Consider a modification to the CRT whereby the overall output is reduced modulo 2^{16} in order to make the physical implementation simpler. The corresponding equation for combining the outputs of the ring generators, assuming $2^{16} > p_i \, \forall i$ becomes:

$$x \mod (2^{16}) = \left(\frac{M}{p_1}\right) \cdot \left[\left(\frac{M}{p_1}\right)^{-1}_{(p_1)} r_1\right]_{(p_1)} + \ldots + \left(\frac{M}{p_n}\right) \cdot \left[\left(\frac{M}{p_n}\right)^{-1}_{(p_n)} r_n\right]_{(p_n)} \mod 2^{16}$$
$$= \left(\left[\left(\frac{M}{p_1}\right) \cdot \left(\frac{M}{p_1}\right)^{-1}_{(p_1)} r_1\right]_{(p_1)}\right)_{(2^{16})} + \ldots + \left(\left[\left(\frac{M}{p_n}\right) \cdot \left(\frac{M}{p_n}\right)^{-1}_{(p_n)} r_n\right]_{(p_n)}\right)_{(2^{16})}$$

A second step of reductions that applies an effective bijective mapping to the output of each residue space is simply removing the inverse multiplication term, which compensates for the rotation occuring when multiplying by $\frac{M}{p_i}$. The multiplicative inverse of the i^{th} ring generator is of the form

$$b_i = \left(\frac{M}{p_i}\right)^{-1} \mod p_i = \left(\prod_{j \neq i} p_j\right)^{-1} \mod p_i = \prod_{j \neq i} \left(p_j^{-1} \mod p_i\right) \mod p_i$$

Since each of the ring generator characteristics are necessarily mutually prime, each p_j has a unique nonzero inverse modulo p_i. Therefore, this multiplication step by b_i simply rotates the i^{th} ring generator output to pre-compensate for the subsequent multiplication by $\prod p_j$, $j \neq i$. The fidelity of the CRT is destroyed if these terms are removed, yet a static multiplication by a nonzero constant in any prime Galois Field is effectively just a bijective mapping between elements in the set. Each residue space is operated upon independently, so the combined output will also have an effective bijective mapping between elements that can be decomposed by each residue space characteristic. Therefore, the deterministic chaotic nature of the output is retained. If the bijective mapping is $g(x)$, then

$$g(x) \mod 2^{16} = \left[\left(\frac{M}{p_1}\right)_{(2^{16})} \cdot (r_1)_{(p_1)}\right]_{(2^{16})} + \ldots + \left[\left(\frac{M}{p_n}\right)_{(2^{16})} \cdot (r_n)_{(p_n)}\right]_{(2^{16})} \mod 2^{16}$$

The reduction of the large multiplication terms and removal of the multiplicative inverse operators provides a significant hardware reduction for an equivalent statistical behavior and a more secure output. A notional block diagram of this hardware efficient modified CRT construction is depicted in Figure 11.

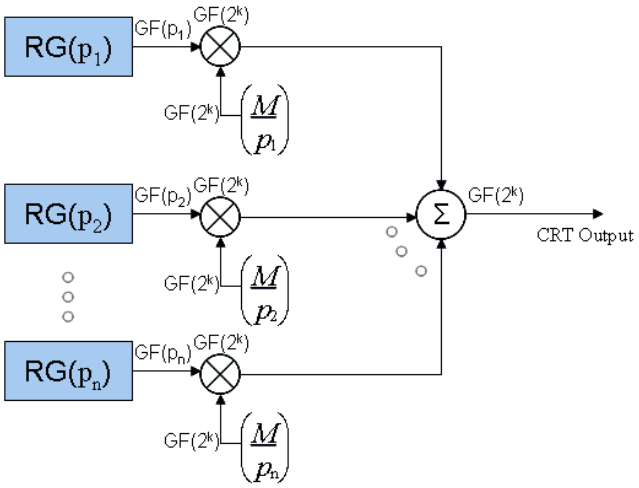

FIGURE 11. Block diagram of modified CRT combination.

2.3.2 Extending Digital Chaotic Sequences

One of the greatest benefits of constructing a random sequence generator in residue number systems rather than weighted number systems is the independence of the hardware implementation and control: there is no need for the results of one computation to be based on another residue limiting the possible number of operations. Further, this methodology provides an efficient mechanism for creating larger dynamic ranges. Considering the modified CRT combination technique, a linear increase in the number of ring generators, and therefore a linear increase in hardware, results in a multiplicative increase in the dynamic range of the effective sequence. Therefore, an efficient, if not too inventive, method for increasing the dynamic range of a discrete-amplitude discrete-time chaotic system is to simply add more ring generators (increase n).

More creative, and inherently more efficient, methods exist, with four distinct options provided in this section as examples[93, 94, 95, 96, 97, 98, 99, 100]. One of the important characteristics that is exploited in the statistical methods is that a uniform distribution on $GF(M)$ will necessarily be non-uniform when reduced modulo 2^k since $\frac{M}{2}$ is not an integer.

► **Permutation-based sequence generation** methods replace the requirements of a chaotic system with simulated randomness based on an external permutation function. In the most extreme sense, the most random that a finite non-repeating sequence (i.e. sampling without replacement) of length p can be is $p!$.

► **Ring generator puncturing** methods take advantage of the characteristics of a mixed-radix conversion between two computational stages by integrating the mixed-radix conversion into the mapping rule. This technique is applied primarily to control the statistical behavior of the sequence.

► **Ring extension multiplication** is available to combine two or more sequences into a single masked sequence. Some engineering liberty is taken in the interpretation of extending the ring $\mathbb{Z}_{p^k}$ even though the proposed method does not satisfy the multiplicative inverse requirements of a field.

► **Mixed-radix accumulation** applies techniques similar to infinite impulse response (IIR) filters to intentionally induce errors in the residue space values that are difficult to recover without knowledge of additional mapping rules and initial conditions.

2.3.2.1 Permutation-based Sequence Generation

The limitation of any ring generator is that it is finite in size; for computations of an irreducible polynomial over a Galois field of prime characteristic p, the ring generator repeats every p^{th} element. A novel way to expand the effective key space is to create a method of ordered, yet apparently arbitrary, sequencing for the p elements in the the field. A ring generator is completely defined by the current state and the mapping rule, whereas an arbitrary ordering of the same elements has $p!$ possible combinations. As an example, consider the prime $p = 233$; the ring generator yields a dynamic range of approximately 8 bits when exhaustive mappings are implemented, whereas an arbitrary ordering would produce a 1501-bit dynamic range.

A small-scale demonstration of this is evident in an arbitrary permutation of 8 input bits of a byte, as might be used in a shift/sift operator. There exist $8! = 40320$ possible permutations of the 8 bits. Using the fundamental theorem of arithmetic, we may rewrite the possible orderings from 8! to $8 \cdot 7 \cdot 6... \cdot 1$ and finally as $2^7 3^2 5^1 7^1$. Viewing the process of creating the unique permutation order as first picking 1 of 8 for the first element, then 1 of 7 for the second element, and so on,[13] we may extract the mixed radix results to uniquely and independently assign a permutation. Now consider an input, $x \mod 8!$, having the decomposition $b_1 b_2 b_3 b_4 b_5 b_6 b_7 t_1 t_2 p_1 s_1$, where b_k are binary digits, t_k are the tertiary digits, p_1 is the pent-iary digit ($x \mod 5$), and s_1 is the sept-iary digit ($x \mod 7$). The first element, choice of 1 from 8, is picked using the first 3 of the 7 binary bits ($2^3 = 8$). The second element is chosen using the remainder modulo 7. The third element is chosen using one binary bit and one tertiary bit (via a miniature CRT combination to create a single number modulo 6). Subsequent bits are

[13]The simplest mathematical interpretation is $8! = 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = \binom{8}{1} \cdot \binom{7}{1} \cdot \binom{6}{1} \cdot \binom{5}{1} \cdot \binom{4}{1} \cdot \binom{3}{1} \cdot \binom{2}{1} \cdot \binom{1}{1}$.

used similarly; the final choice determined by the independent sequences

$$b_1 b_2 b_3 \qquad s_1 \qquad b_4 t_1 \qquad p_1 \qquad b_5 b_6 \qquad t_2 \qquad b_7$$

with all 40320 permutations made possible. The generalized permutation case is as shown in Figure 12.

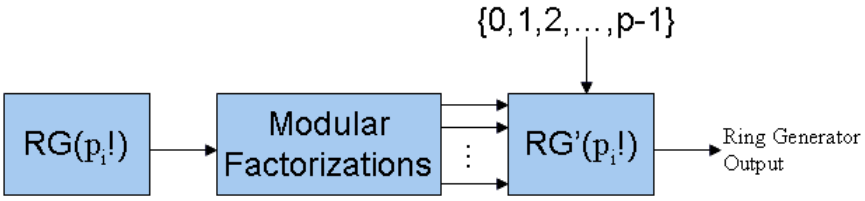

FIGURE 12. Block diagram of condensed permutation mapping.

The modified generator converts a random number generator taken over a key space of $p!$ as the unique permutation ordering for the next p output elements; its output values will not repeat for $p \cdot p!$ outputs, making unintended reconstruction of the sequence more difficult.

Application of this technique requires additional care – only two such generators may be combined in parallel (as in the CRT or modified combination algorithm) if the characteristics are within a factor of two. Consider two modified generators, mapped to two distinct prime characteristics, p and q, with $p > q$. The first modified generator will repeat every $p \cdot p!$ elements, while the second modified generator will repeat every $q \cdot q!$ elements. Since $p > q$, $\frac{p!}{q!}$ is an integer. Moreover, the ratio $\frac{p \cdot p!}{q \cdot q!}$ is an integer if $p > 2 \cdot q$, and a non-integer if $p < 2 \cdot q$, since q only divides $p \cdot \prod_{u=q+1}^{p} u$ when some u is an integer multiple of q. The combined sequence length for two generators is therefore

$$\text{Sequence Length:} \quad \begin{cases} p \cdot q \cdot p! & q < p < 2q \\ p \cdot p! & 2q < p \end{cases}$$

This technique is only useful for a single ring generator unless combined with different techniques that are based on inputs from other number radices. This method becomes more practically useful when an external pseudorandom source provides a periodically updated seed that is mapped to the generalized permutation ordering or a relatively prime clocking/enable structure is implemented in hardware.

2.3.2.2 Punctured Ring Generators

The non-uniform distribution that results from the mixed-radix ring generator structure is in itself non-ideal: if only a small number of outputs are used (say $\frac{p}{100}$) and the ratio of

prime to final field characteristic ($\frac{p}{2^k}$) is large, then this non-uniformity may be negligible. The singular partial cycle will be sufficiently masked by the pseudo-randomness of the remaining full cycles. As the ratio of prime to final field characteristic gets smaller, the non-uniformity becomes more relevant. One solution to this is to intentionally reduce the size of the prime field by puncturing the cycle as necessary.

Consider again a simple mixed-radix system where $p_1 = 233$ and $p_2 = 5$. Some type of pseudo-random mapping, f, is created over $\mathbb{Z}_{233}$ and then reduced modulo 5. There will be $\lfloor \frac{233}{5} \rfloor = 46$ complete cycles through $\mathbb{Z}_5$ and one partial cycle containing 3 additional elements. The prime mapping should then be punctured strategically at 3 distinct points (one each in the equivalence classes of $\mathbb{Z}_{233}$ that map to 0, 1, and 2 modulo 5) such that $f(f^{-1}(x)) = f(x)$. A notional diagram of a punctured ring transforming from $\mathbb{Z}_{2^k}$ to $\mathbb{Z}_p$ is shown in Figure 13.

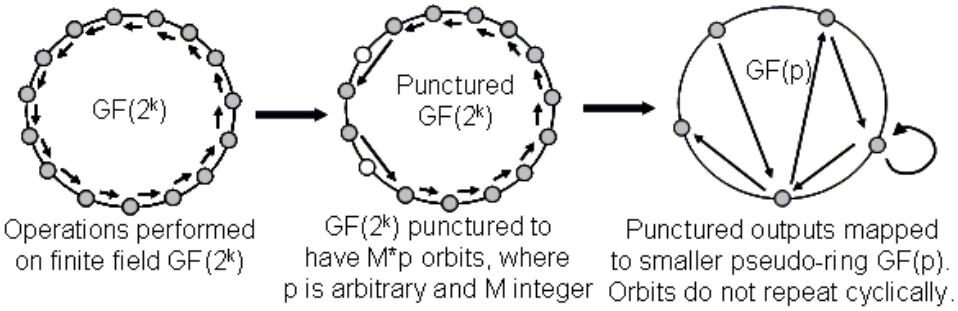

FIGURE 13. Puncturing ring generators to eliminate statistical artifacts.

That is, the element in $\mathbb{Z}_{233}$ that is just before x in the $\mathbb{Z}_{233}$ cycle, $f^{-1}(x)$, has its mapping replaced by the next element after x, effectively skipping x in the cycle. Statistically, this ensures a perfectly uniform distribution over $\mathbb{Z}_5$. Additional considerations will need to be made if this technique is used with combined ring generators (the modified ratio $\lfloor \frac{233}{5} \rfloor = 46 = 2 \cdot 23$ becomes an issue in assessing mutually prime combination), yet can be avoided if the combination is performed prior to puncturing. This technique is analogous to puncturing an error-correction code in communication systems or stabilizing unstable periodic points in one-dimensional nonlinear systems, hence the chosen name[101].

2.3.2.3 Ring Extension Sequence Combination

Another method for extending the length of a random sequence is to weave it together with another sequence that is of a mutually prime characteristic. This general concept is used in Gold code generators[102] and also in generation of complex spreading sequences like the Global Positioning Satelite P-code[103]. Generalizing this technique to something other than $GF(2)$ provides an effective way to mask one sequence with another and/or combine two

sequences in an invertible manner that inherently contains a degree of security from reverse engineering. That security translated into additional users in multiple access communications and optional permission-based physical layer separation.

Starting with the idea of sequence combination in $GF(2)$, an addition operator may be efficiently implemented using an XOR logic gate. Each input is a one-bit binary value, where the rollover characteristic of $1 + 1 = 2 \equiv 0$ is identical to the output for $0 + 0 = 0 \equiv 0$. From the viewpoint of extending the sequence length, combining two ring generators with mutually prime periods T_1 and T_2 will necessarily result in a sequence of length T_1T_2 since the independent epochs of the length-T_1 ring generator and length-T_2 ring generator fail to occur simultaneously. The partial probability that the second input sequence was a particular value given the first input sequence is identically equal to the unconditional probability of $\frac{1}{2}$. That is, no a priori information of the sequence input lengths is provided without extensive historical knowledge of the sequence outputs and the mapping rule. Numerous methods exist for reverse engineering linear feedback shift-register constructions[104, 105, 106], but the general principle holds that the larger number of distinct sequences that are combined, the more difficult the sequence is to unravel.

This proposed technique will work for any p-adic sequence (eliminating the authentication problem of the simplistic example) or as a dynamic range expansion and sequence combination technique as shown for a cryptographic application in Figure 14[95, 100]. Although not entirely correct in satisfying all the axioms of a finite field, the arithmetic relies on operators that are commonly called GF multiplies (requires approximately half the gates that a full hardware mulitiplier needs), GF adders, etc.[107], so is discussed in this dissertation as such; handling of the multiplicative inverse during discussion of sequence inversion.

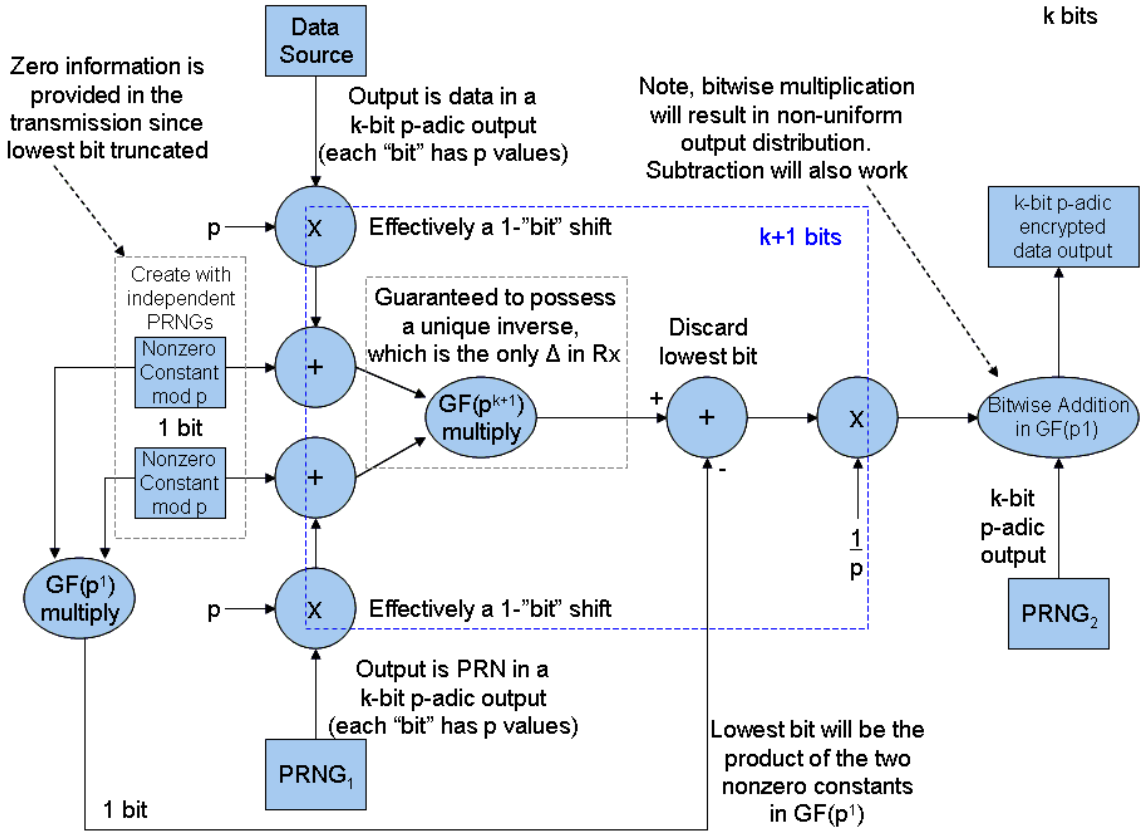

FIGURE 14. Block diagram of p-adic sequence combiner.

The "closed" property of a GF system is obtained by performing an affine transformation on each input prior to multiplication in a Galois Field one order higher than the inputs. The transformation and overall operation for a pair of k-bit p-adic inputs, x_1 and x_2, is:

$$\frac{\left[(p \cdot x_1 + c_1) \cdot (p \cdot x_2 + c_2) \mod p^{k+1}\right] - (c_1 \cdot c_2 \mod p)}{p}$$

The two constants c_1 and c_2 must be nonzero elements in $GF(p)$, which greatly contribute to the overall result, and ensure that the output is both uniformly distributed and reconstructable by similar multiplication by an inverse element modulo p. These constants may be chosen as an independent random sequence in and of themselves, further increasing the possible sequence length. Choosing $p = 2$ reduces this model to the usual case of a binary number system and further necessitates that the nonzero elements be 1. The complete set of transformations for a pair of k-bit binary numbers x_1 and x_2, where x_1 is the unknown message and x_2 is taken from a chaotic sequence reduce to

$$\frac{\left[(2 \cdot x_1 + 1) \cdot (2 \cdot x_2 + 1) \mod 2^{k+1}\right] - 1}{2}$$

It is trivial to see that every odd number has a unique inverse within $GF(2^{k+1})$ for any k, while no even number has an inverse in $GF(2^{k+1})$ (simply prove that 2 has no inverse). As a

result, there exist 2^k elements with unique inverses in $GF(2^{k+1})$. Repeating this process at the recipient end, we see the original message is readily obtained; the simple calculations below demonstrate x_1 being combined with x_2 and then re-obtained via x_2^{-1}.

$$\frac{\left(2\frac{((2x_1+1)\cdot(2x_2+1))-1}{2}+1\right)(2x_2+1)^{-1}-1}{2} = \frac{(2x_1+1)(2x_2+1)(2x_2+1)^{-1}-1}{2} = \frac{(2x_1+1)\cdot 1-1}{2} = x_1$$

Extending this discussion back to the general case of $GF(p^k)$, the transformations presented are used to perform a multiplication of two elements in $GF(p^k)$ within the extension field $GF(p^{k+1})$ and ensure a unique inverse element that may be used to reverse the process. The output distribution is uniform (measured on $GF(p^k)$) whenever one of the inputs is uniformly distributed; we have assumed implicitly that both inputs are independent. As a result, combining two independent inputs with uniform distributions over mutually prime key spaces will produce a uniform output over the product of the input spaces – an extremely efficient operation considering the Galois field multiplication is equivalent to a $(k+1)$-bit binary multiplication with all bits higher than 2^k discarded. In the simplest sense, inputs and outputs are related via a bijection for the binary case; the mapping become more secure when performed over an RNS-based domain[108].

2.3.2.4 Mixed-Radix Accumulation

A final exemplary method to remove the statistical artifacts, and one that is significantly easier to implement, is to adapt the hardware to spread mixed-radix conversion non-uniformities about the destination ring. In a highly parallel system where $\frac{M}{2^k}$ is sufficiently large, the excess elements in each individual residue space may be independently accumulated so that their average occurrence spreads throughout the number base. Provided all numbers are mutually prime, this operation produces a random rotation in each number base; when the numbers are not mutually prime (leading to fixed points on average), a fixed additional rotation may be induced to eliminate fixed points. Further, the accumulator structure may be used to inject another state altering initial condition, yielding more degrees of freedom in defining the initial state, or equivalently a larger key space. Further still, this structure provides an ideal interface to induce pseudorandom (yet coherent) errors into the system that prevent long-term acquisition of the output[108]. A block diagram of a mixed-radix accumulator that converts inputs from $GF(M)$ to $GF(P)$, where $P = \prod_{i=1}^{k} p_i$ is mutually prime with M and $H(z)$ denotes any chosen filter structure whose arithmetic is constrained to $GF(P)$.

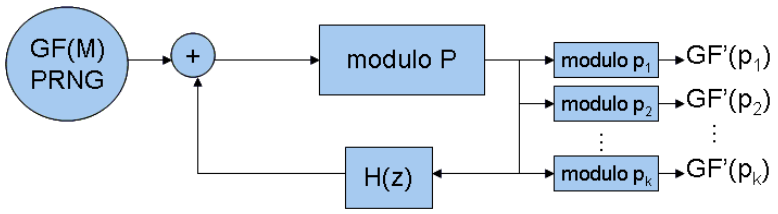

FIGURE 15. A mixed-radix accumulator to eliminate statistical artifacts.

2.3.3 Prototype Digital Chaotic Circuit

The end goal of constructing a digital chaotic circuit for a coherent communication system is not necessarily to have the closest analogy to a continuous chaotic system that hardware can provide, but rather a measurably indistinguishable solution that demonstrates all of the properties of a chaotic sequence applied to a communications waveform. Subsequent work has shown that any good discrete-amplitude discrete-time random number generator may be used for "chaotic" processing; the benefit of retaining the firm basis in irreducible polynomial computations followed by modified CRT combination is that the statistical properties and extremely long sequence length are mathematically assured. Further, the control mechanisms for synchronizing two independent digital chaotic circuits are simplified by continuing the mixed-radix representation of the system state.

This section discusses various options for implementating the chaotic sequence processing efficiently in hardware. Trades between hardware utilization, processing latency, and control mechanisms dominate the discussion. At the conclusion of this section, an exemplary structure for a producing practically infinite length sequences is provided; application of this prototype sequence generator to a chaotic communication system is provided in Chapter 3.

2.3.3.1 Efficient Ring Generator Implementation

As the core computational units in this mixed-radix sequence generator, the efficiency of the ring generator design will be the most dominant hardware utilization driver. The easiest analytical solutions to implementing the ring generators are also the least efficient: tailored binary hardware for implementing the computation in each individual residue space; a notional hard-coded architecture for computing $f_{11}(x) = 3x^3 + 3x^2 + x + 7$ is depicted in Figure 16.

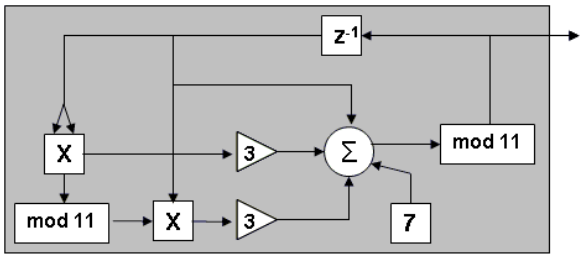

FIGURE 16. A brute force ring generator implementation of $f_{11}(x) = 3x^3 + 3x^2 + x + 7$.

Both the use of hardware multipliers and repeated modular reductions would result in a wasteful use of hardware; modifications can be made for small primes like $p = 11$, yet fail to provide a generalized structure for larger primes. It is worth noting that the exact same computation is performed each cycle, and moreover, the finite domain and range of the mapping can be exploited by pre-computing the function evaluations and then calling them as indices out of a static ROM. An improved implementation that employs lookup tables (LUT) for the chaotic polynomial evaluations is depicted in Figure 17.

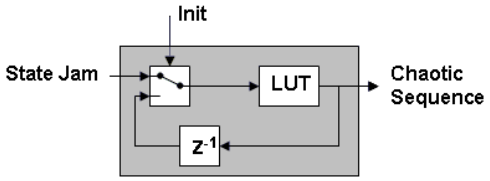

FIGURE 17. A table-based ring generator implementation of $f(x)$.

A table-based polynomial implementation removes any concern of the order of the polynomial since all values are pre-computed. The values stored in the i^{th} table take the form

$$\text{LUT Entries:} \quad \{f_{p_i}(0), f_{p_i}^{(2)}(0), \ldots, f_{p_i}^{(p_i-1)}(0)\}$$

Further, the structure lends nicely to a repeated mapping since the next input is the current output. The *state jam* input to this ring generator is a method to insert an initial state into the generator before enabling. An improved version of the ring generator recognizes that there must be controls to vary timing, reset, and synchronize the sequence with another copy. So far, the implementation reduction has converted all real-time, non-binary arithmetic into simple LUTs; this method is efficient provided the prime residue is not too large ($< 2^{11}$).

2.3.3.2 Efficient Modified CRT Combination

Similar to the reduction of arithmetic in residue spaces, the modified CRT combination of the ring generators may be implemented using a table-based approach rather than hardware multipliers. As a result, the chaotic sequence generator may be constructed quickly using $2n$ LUTs, n unit delays, and a single binary adder. The LUT sizes are defined by the prime residues p_i. A notional hardware implementation for the chaotic sequence generator $f^*_{M=3,563,762,191,059,523}(x) = 3x^3 + 3x^2 + x + 97,903,550,178,815$ shown in Figure 18

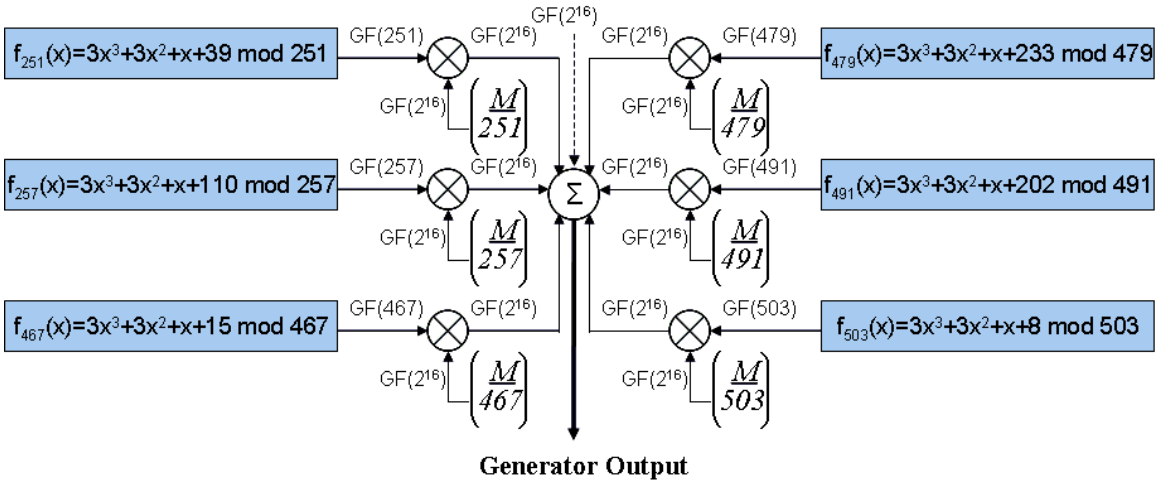

FIGURE 18. Prototype chaotic sequence generator.

is depicted in Figure 19[109]. The mappings in the second set of tables are 16-bit (modulo 2^{16}) remainders of the multiplication of $(\prod_{j \neq i} p_j)$ and the ring generator output r_i.

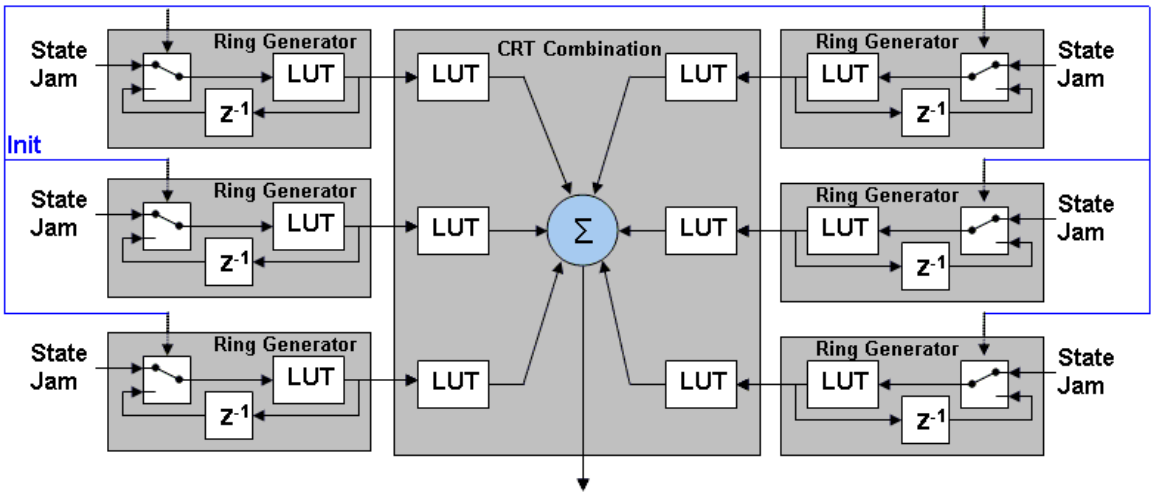

FIGURE 19. A table-based chaotic sequence generator.

Note that the first LUT containing the chaotic polynomial mapping may actually be integrated together with the second (bijective) mapping for modified mixed-radix combination.

This composite mapping takes the form

$$\text{LUT Entries:} \quad \{\left(f_{p_i}(0)\cdot\frac{M}{p_i}\right)_{(2^{16})},\left(f_{p_i}^{(2)}(0)\cdot\frac{M}{p_i}\right)_{(2^{16})},\ldots,\left(f_{p_i}^{(p_i-1)}(0)\cdot\frac{M}{p_i}\right)_{(2^{16})}\}$$

and may be strictly implemented in a single LUT, provided a mechanism exists for keeping track of the residue from one cycle to another. Note that the periodicity of this sequence has been used to convert a moderately random process to a time-indexed process where choice of the topologically transitive mapping (such as an irreducible chaotic polynomial) ensures that for any chosen initial condition $y \in GF(p_i)$

$$f_{p_i}^{(p_i)}(y) = f_{p_i}^{(0)}(y) = y \qquad \text{and} \qquad f_{p_i}^{(k)}(y) \neq y \quad \forall\ 1 \leq k \leq p_i$$

The LUT may be based solely on the initial condition and the time index k.

2.3.3.3 Control Mechanisms

To construct a robust chaotic sequence suitable for a practical communications system, he evolution of the chaotic sequence must be flexibly controllable. This is a key difference between a discrete-amplitude discrete-time chaotic circuit and a notional continuous (or even analog) chaotic circuit, building in the hooks to initialize, synchronize, and maintain steady state lock between multiple digital chaotic circuits that depend on inherently different physical processes (different clock references, motion/Doppler characteristics, RF hardware, etc.). The key mechanisms are a method to immediately jump to any time index in the lifetime of the sequence, quickly advance or retard a chaotic sequence value by any number of cycles (integer and fractional chip periods for optimal lock) during acquisition, and dither the chaotic sequence by fractional chip periods (accounting for chip time boundaries) to respond to time tracking. Since each of the ring generators operate independently, these control mechanisms must also guarantee that the ring outputs maintain relative synchronization with one another during any command. A modified block diagram implementing the desired controls is shown in Figure 20[110].

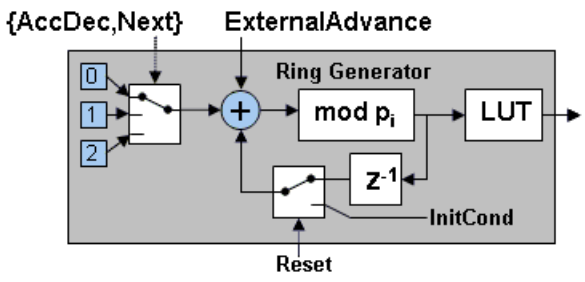

FIGURE 20. Digital chaotic sequence ring generator with controls.

During initialization, the ring generator is cleared of its current residue/state by strobing the boolean *Reset*, exchanging the accumulated time index value for a chosen initial condition *InitCond*; the next output of the chaotic sequence is then $f_{p_i}(InitCond)$, where *InitCond* is assumed to have been previously reduced modulo p_i. A pair of controls *AccDec* and *Next* provide the ability to increment the sequence time index by zero ($(AccDec, Next) = (1, 1)$), one ($(AccDec, Next) = (x, 0)$), or two ($(AccDec, Next) = (0, 1)$) cycles, corresponding to retarding the sequence (physical time elapses, but the sequence stays put), evolving the sequence, and accelerating the sequence (effective sequence evolution is twice the standard), respectively. The *ExternalAdvance* control is a jump control that permits a static time jump through the sequence at any time; the input value is assumed to be an element of $GF(p_i)$. The time increment selection, external advance, and previous time value are all added together (adder is $\lceil \log_2 p_i \rceil + 1$ bits wide) followed by a hardcoded modular reduction mod p_i. The largest instantaneous value that will be produced by the adder is $2p_i - 1$.[14] After the modular reduction, the maximum index value that will be inserted into the LUT (containing the 16-bit output representing the $\mod 2^{16}$ reduced $\frac{M}{p_i} f_{p_i}^{(k)}(0)$ evaluation) will be $p_i - 1$, which is consistent with a view of the chaotic polynomial being evaluated on $GF(p_i)$. Note that the result of the polynomial computation (the residue) is never actually calculated in any intermediate steps.

In steady-state operation, the ring generator time index k will increment and reset similar to a prime-valued counter.

$$k_{\text{Normal Operations}}: \qquad 0 \rightarrow 1 \rightarrow 2 \rightarrow \ldots \rightarrow (p_i - 1) \rightarrow 0 \rightarrow \ldots$$

The only time the additional controls are used is during chaotic state initialization, acqisition, synchronization, or receiver signal tracking.

2.3.3.4 Non-Chaotic Masking Sequence

A second set of sequences that use non-chaotic, yet exhaustive, mapping techniques and also apply the tricks of Section 2.3.2 was constructed to assist in masking the digital chaotic sequence from reverse engineering. The chosen method employs a distinct set of primes $\{q_i\}$ from those used in the chaotic ring generators for the secondary mapping, with the additional constraint that pairs of primes add to a power of two. These modified ring generators are added selectively (the output of a q_i-based ring generator with an output of its complementary size $q_{i*} = 2^r - q_i$ ring generator) to force the statistical aliasing of the uniform distribution caused by mixed-radix combination to create a near-uniform distribution. An example of such

[14] The accumulated value and external advance can each be as high as $(p_i - 1)$, added to a time increment of 2, for $2(p_i - 1) + 2 = 2p_i$. The external control will prevent the value from ever exceeding $2p_i - 1$.

a generator is $(q_i, q_{i*}) = (73, 439)$, which produces the following statistical distributions when (a) considered independently in residue spaces, (b) reduced independently modulo 2^8, or (c) combined modulo 2^8 for a single $q_i \cdot q_{i*}$ cycle as shown in Figure 21 ($q_i = 73$ on top).

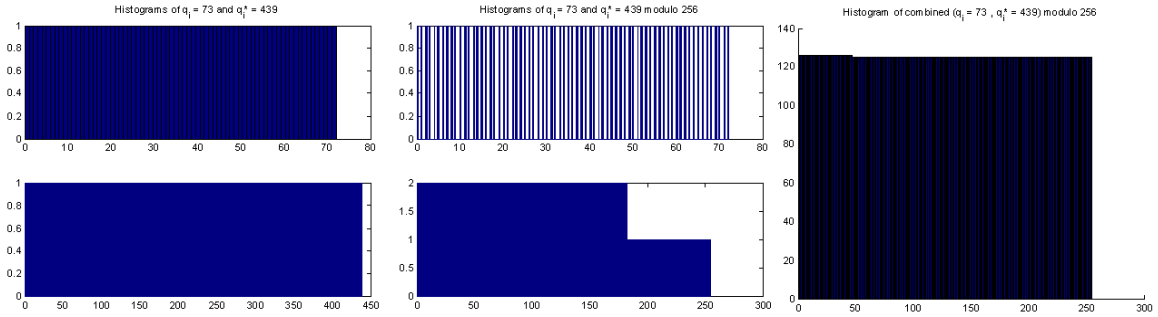

FIGURE 21. Histograms of masking sequence components.

This masking sequence was constructed using a similar topology and control structure as the digital chaotic circuit, adding a modulo 2^8 addition at the paired outputs as shown in Figure 22. This sequence is enabled at one-half the period of the chaotic sequence, which is broken into successive samples for computation of in-phase and quadrature components.[15]

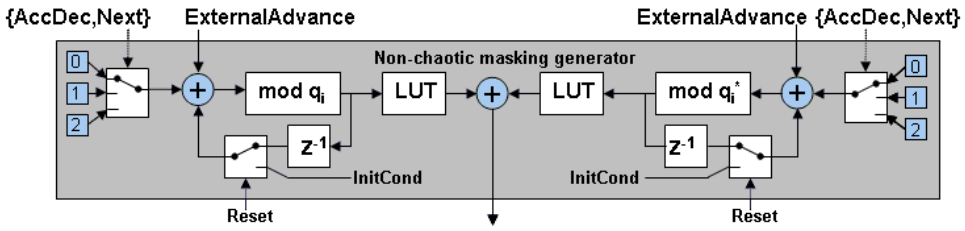

FIGURE 22. Block diagram of masking sequence generator.

2.3.3.5 Box-Muller Statistical Transformation

An assumption throughout the construction of these digital chaotic circuits is that the resulting signal will approximate additive white Gaussian noise. The resulting signal should take the form of a quadrature (single-sideband) pair of Gaussian random variables with equal power spectral density over the entirety of its bandwidth; at any given point in time, the complex signal has a Rayleigh distributed magnitude and a uniformly random phase. The RNS-based digital chaotic sequence, however, most nearly approximates uniformly distributed random numbers, which may be transformed into the desired quadrature pair of Gaussian random variables via the Box-Muller transformation[111, 112, 113].

[15]Another method that may be used to increase the effective dynamic range (repetition period) of the chaotic sequence is to use a mutually prime divisor in the clocking rate of the various ring generators; this method does however require additional control complexity.

The Box-Muller transformation implements the bivariate mapping

$$X_I = \sqrt{-2\sigma_x^2 \log u_1}\cos(2\pi u_2) \qquad X_Q = \sqrt{-2\sigma_x^2 \log u_1}\sin(2\pi u_2)$$

where $u_1$and u_2 are uniformly distributed random numbers on [0, 1) created through the chaotic sequence generator. Based on the analytical and measured performance of the chaotic sequence generator, u_1 and u_2 may either be successive outputs of the same chaotic sequence generator or, preferably, developed from two independent chaotic sequence generators constructed with mutually prime ring generators; i.e. $gcd(M_1, M_2) \equiv 1$.[16] This nonlinear mapping creates a quadrature pair of Gaussian distributed random variables[17] where the magnitude of X, calculated as

$$|X|^2 = (X_I^2 + X_Q^2) = \sqrt{-2\sigma_x^2 \log u_1}^2 (\cos^2 2\pi u_2 + \sin^2 2\pi u_2) = -2\sigma_x^2 \log u_1$$

has a Rayleigh distribution, identical to a quadrature measurement on AWGN. Implementing this mapping efficiently in hardware, quick observation shows a drastic nonlinearity in the mapping. The magnitude mapping, $\sqrt{-2\sigma_x^2 \log u_1}$, is shown in Figure 23 for $\sigma_x = 1$ and $0.0001 < x < 0.9999$. Note that the mapping is divergent at its lower endpoint; that divergence is equivalent to a Gaussian distribution having theoretically infinite tails.

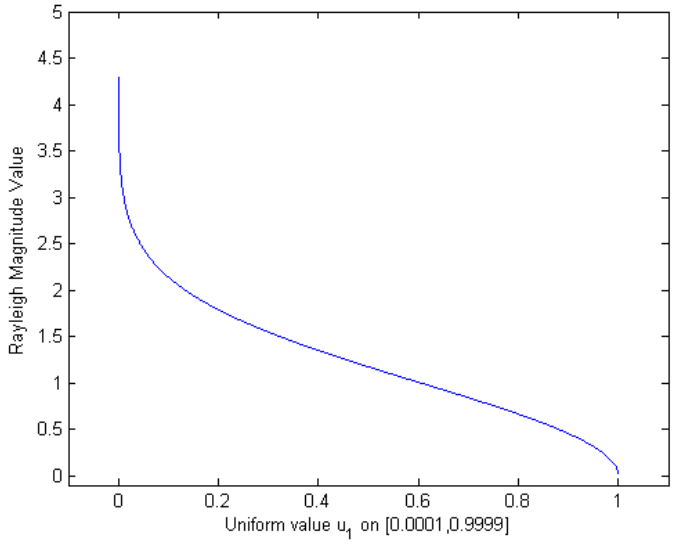

FIGURE 23. Rayleigh magnitude mapping for the Box Muller transformation.

[16]Using two chaotic sequence generators with mutually prime characteristics will make the effective repetition period of the bivariate Gaussian mapping $M_1 \cdot M_2$, increasing sequence length; a single generator with decommutated outputs was chosen for the prototype chaotic communications system to reduce hardware utilization.
[17]"Quadrature" is ensured by the orthogonal mapping of the sine and cosine operators, while the randomness is inherited from the uniformly distributed chaotic sequence generator.

The sine and cosine components of the Box-Muller transformation are more straightforward, with the primary concern being implementation efficiency; this section will discuss one specific design for an efficient hardware sine/cosine generator based on nonlinear processors[114], although various generalizations have been identified and selected for patent protection[115].

2.3.3.5.1 Efficient Implementation of Rayleigh Magnitude Calculator

To determine the precision needed in constructing the nonlinear $\sqrt{-2\sigma_x^2 \log u_1}$ mapping, we need only to meet the dynamic range of the transmitter output stage, which will contain a digital-to-analog (D/A) converter with 12 to 16 bits of effective precision. We also note that the peak-to-average-power ratio (PAPR) will be set in this stage, based on the determination of the maximum allowable magnitude in the standard normal distribution; in theory, the standard normal distribution has infinite tails, while in practice the outlier values that are significantly outside the notional $\pm 3\sigma$ range rarely occur. The PAPR measured in dB, assuming symmetric truncation, is equivalent to

$$PAPR = 20 \log_{10} \frac{|\sigma_{\text{Trunc}}|}{\sigma_x} = 20 \log_{10} |\sigma_{\text{Trunc}}|$$

where σ_{Trunc} is the number of standard deviations out from the mean that are retained by the mapping. All input values exceeding σ_{Trunc} are either discarded (simple conceptually, but difficult in implementation) or saturated to a maximum value of σ_{Trunc} (simple in implementation, yet results in a noticeably non-Gaussian distribution). Truncating the normal distribution too close in either case distorts the distribution by inducing measurable waveform features that reduce the signal entropy. Consider the histogram of 1,000,000 random draws from Matlab's standard normal random number generator depicted in Figure 24 that shows a range of possible values at which to saturate the distribution.

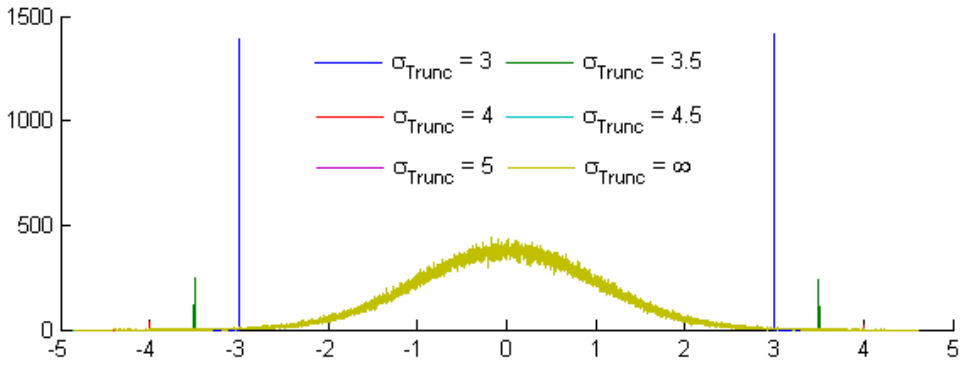

FIGURE 24. Truncation effects for standard normal distribution.

As would be expected, saturating the tails of the distribution result in point masses at $\pm\sigma_{\text{Trunc}}$ equal in mass to the integrated decaying tail. For example, truncating this random selection at $\pm 3\sigma$ results in point masses having approximately 0.140% of the total mass compared to the expected value of 0.135% of a perfect Gaussian distribution. Further, these point masses slightly distort the higher-order statistical characteristics, with a sample skewness of 0.0052 and kurtosis of 2.922 at $\sigma_{\text{Trunc}} = 3\sigma$; truncating the sequence as far as $\sigma_{\text{Trunc}} = 2.5\sigma$ results in a kurtosis of 2.758, which is significantly different than 3. To demonstrate this degradation in the kurtosis and other cumulants more clearly, a plot of the cumulant versus truncation limits is shown in Figure 25; the odd order cumulants are plotted directly, while the even order cumulants are plotted as fractional values.[18]

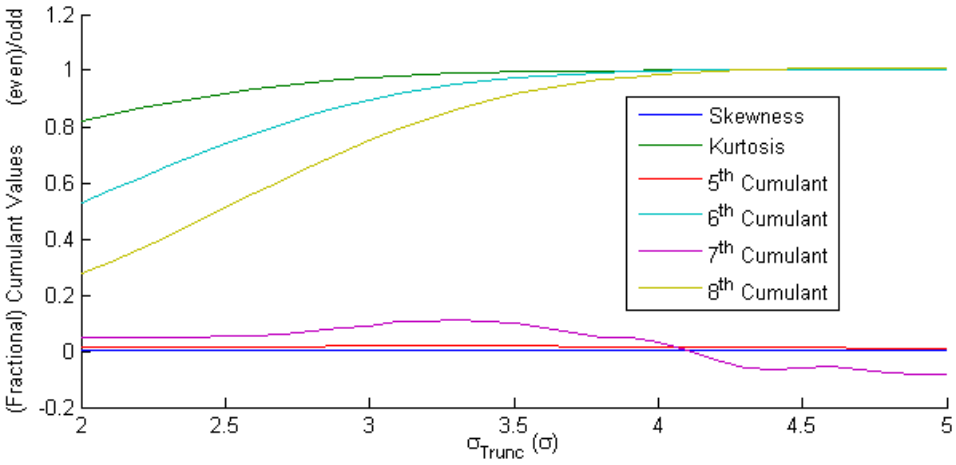

FIGURE 25. Range of cumulant values for truncated normal distributions.

Note that the non-Gaussian characteristic is exceedingly easy to determine as higher order cumulants are calculated; it is expected that cumulants above 6^{th}-order are probably too difficult to implement reliably, yet choose to construct the Box-Muller transformation based on a value of $\sigma_{\text{Trunc}} \approx 4.3\sigma$, which equates to a relatively high PAPR of 13.1 dB.[19] A second consideration in the chaotic waveform development is whether the rounding/truncation errors that are induced by finite precision arithmetic adversely impact the cumulant evaluations. Using a Matlab simulation to quantify this finite-precision impact to the cumulants over 1M standard normal samples results in the cumulant curves shown in Figure 26; it is assumed that the cumulant calculations are processed with infinite precision and that the NLP values are either truncated (o) or rounded (x).

[18]The ideal kurtosis is 3, ideal 6^{th}-order cumulant is 15, and the ideal 8^{th}-order cumulant is 105.

[19]The effects of HPA compression may actually be mitigated via predistortion at this stage using similar techniques to multi-carrier or CDMA systems[116, 117].

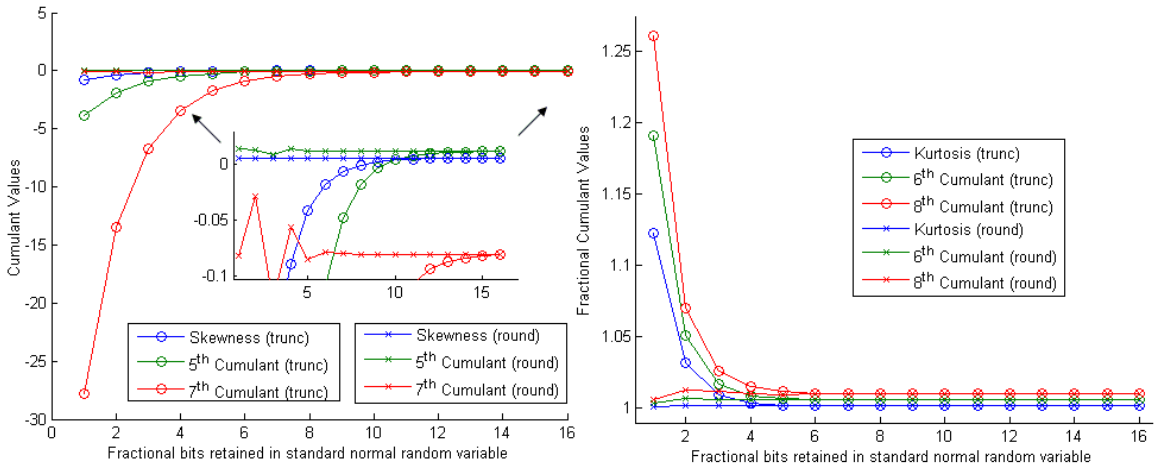

FIGURE 26. Impacts of finite precision NLP arithmetic on cumulants.

Clearly, the higher-order odd comulants are affected more by the finite precision arithmetic, with truncation tending to make the non-centered LSB in twos complement binary representation increasingly significant. Rounding is much preferred to truncation for this reason. As a result of these plots, it appears that any standard normal random variable with at least 8 fractional bits and rounding LSB decisions will retain the desired statistical characteristics. The developed NLP is precise to approximately 14.5 bits of precision, well above this minimum requirement. Evaluating the nonlinear mapping for converting a 16-bit uniform random value on [0,1) to a Rayleigh magnitude more closely, we see that extremely small values of the input map to large values in the output and that the mapping is monotonic decreasing over the [0,1) input domain. As stated previously, the mapping is divergent at the lower end.

$$\lim_{u_1 \to 0} \sqrt{-2\sigma_x^2 \log u_1} = \infty \qquad \lim_{u_1 \to 1} \sqrt{-2\sigma_x^2 \log u_1} = 0$$

Rather than implement this mapping using traditional methods like CORDIC functions (iterative and slow), splines (design and computationally intensive), or computer processor (hardware intensive and slow), a hybrid LUT and calculation method called a nonlinear processor (NLP) was designed. NLPs have been used in numerous applications as an efficient means of implementing nonlinear computations in FPGAs and ASICs[118, 119]. The chosen precision truncates the standard Normal distribution at approximately 4.33 standard deviations, which is beyond the resolution of the eventual digital to analog conversion and detectability in the cumulants.[20] A Simulink block diagram of the Rayleigh magnitude mapping is shown in Figure 27. Note that blocks denoted "virtual blocks" are artifacts of the simulation (typically changing from one data type to another) required to most closely mimic hardware implementation.

[20] An input value of 0 was arbitrarily mapped to that of 2^{-17} before computing the nonlinear function.

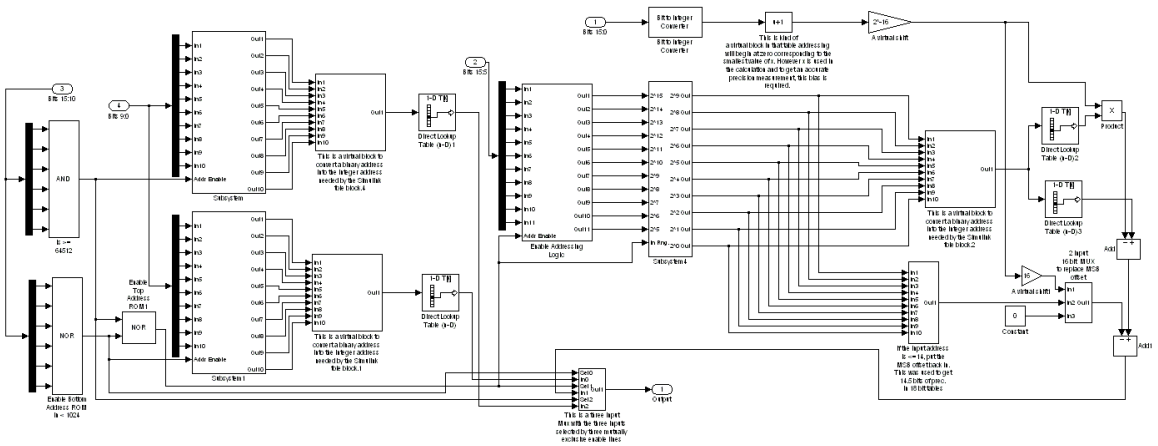

FIGURE 27. Simulink block diagram of Rayleigh magnitude nonlinear processor.

The processing flow is broken up into three primary paths where the two boundary conditions (input values near 0 or 1) are treated precisely with LUTs and other values are computed based on combined Taylor series approximations and advantageously placed shift-add operations. The effective result is approximately 14.5 bits of precision in the nonlinear mapping at a hardware cost of four LUTs and one multiplier. An initialization script is included in the appendix to describe the arithmetic behind the approximations.

2.3.3.5.2 Efficient Quadrature Sine/Cosine Generation

The digital generation of quadrature sine and cosine values of an input angle is a well studied problem, with solutions ranging from recursive CORDIC algorithms, static LUTs, polynomial approximations, and nonlinear processors. The chosen model was a moderate precision ($\approx$ 16-bit) nonlinear processor (NLP) architecture that exploits octant sinusoidal symmetries and implements hybrid LUT/real-time processing for a hardware efficient solution. Similar architectures have been used to construct hardware efficient sine/cosine generators with 24.7 bits of precision[115] and can be extended to even higher precisions for other applications. The specific NLP design for the digital chaotic communications system is shown in Figure 28.

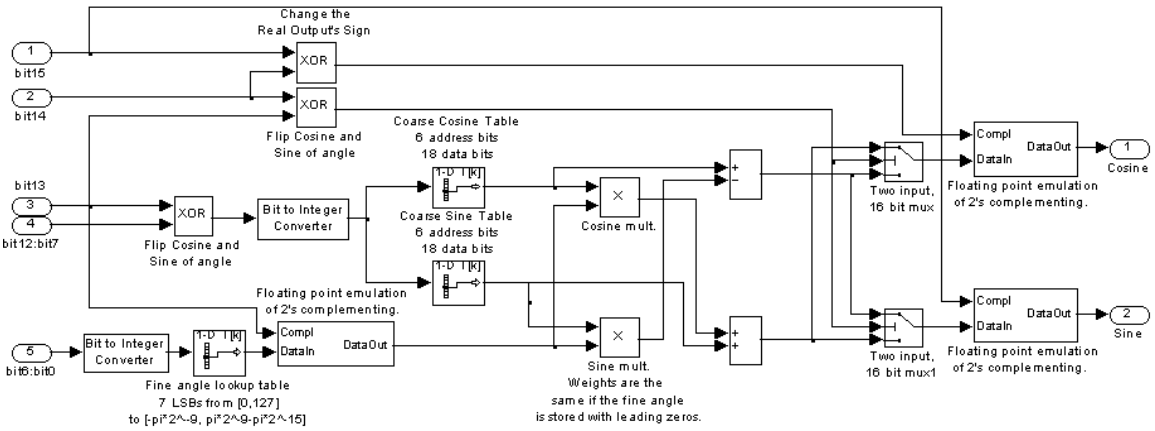

FIGURE 28. Nonlinear processor generation of precise sine and cosine evaluations.

The theory of operation for the mapping is to first determine the output octant (using the three MSBs), and then exploit octant symmetry for the simultaneous sine and cosine mappings. A pair of LUTs (using bits 3:8) are used to determine the coarse mappings within the octants for the sine and cosine with the following rules (in Matlab notation); all input values are taken as integers into the LUTs, with a dynamic range of $[0, 2^6 - 1]$. The rounding operation was performed to limit the number of LUT entry bits to 18 (based on destination FPGA hardware).

$$\text{Cosine}_{\text{Coarse}} = \text{round}\left(2^{17} \cos\left[\frac{\pi}{512} : \frac{\pi}{256} : (\frac{\pi}{4} - \frac{\pi}{512})\right]\right) \cdot 2^{-17}$$

$$\text{Sine}_{\text{Coarse}} = \text{round}\left(2^{17} \sin\left[\frac{\pi}{512} : \frac{\pi}{256} : (\frac{\pi}{4} - \frac{\pi}{512})\right]\right) \cdot 2^{-17}$$

A similar operation is performed in parallel using the 7 LSBs to obtain a fine-grained sine/cosine mapping, according to the following rule. Again, the entries to the LUT are taken as integers on $[0, 2^7 - 1]$ with the 7 LSBs.

$$\text{Sine/Cosine}_{\text{Fine}} = \text{round}\left[\sin\left(\left[-2^6\pi : \pi : (2^6 - 1)\pi\right] \cdot 2^{-15}\right) \cdot 2^{17}\right] \cdot 2^{-17}$$

The coarse- and fine-grained estimates of the sine and cosine values are then combined and modified for the appropriate octant; the overall sine and cosine outputs provide approximately 110dB in amplitude quantization spurious free dynamic range (SFDR) and 92dB of phase accuracy at a hardware cost of 4.6Kb in LUTs and 2 multipliers.

2.3.3.6 Combined Digital Chaotic Sequence Generator

The final design for the digital chaotic sequence generator combined 16 ring generators implementing chaotic polynomial evaluations with 8 dual-prime masking sequence generators for a total dynamic range (sequence repetition period) of $2^{278} \approx 10^{83.7}$, which is practically infinite. A block diagram including all components is shown in Figure 29.

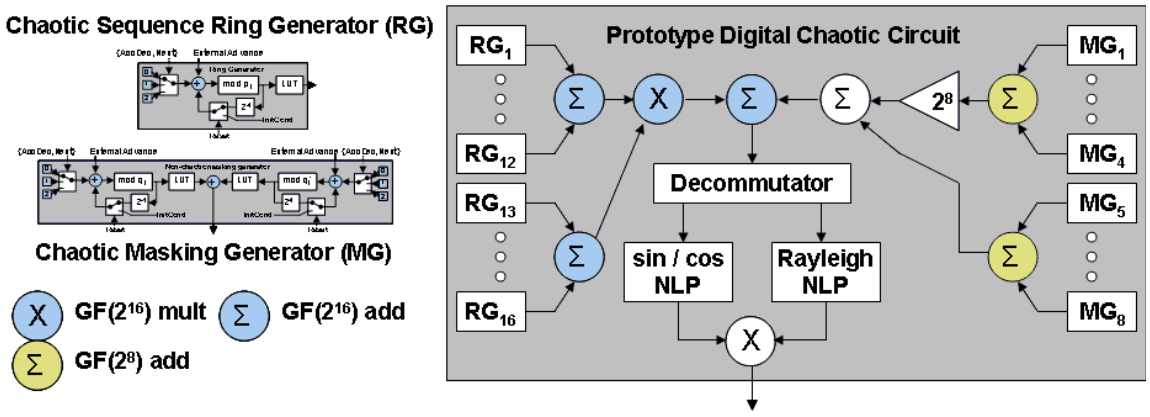

FIGURE 29. Block diagram of prototype chaotic sequence generator.

A series of Matlab initialization scripts is included in the appendix to describe the set of prime values chosen for each of the chaotic sequence ring generator characteristics, construction of the chaotic irreducible polynomials, instantiation of the masking sequence generators, nonlinear processor coefficients, and all lookup table values.

2.3.3.7 Hardware Utilization

The digital chaotic sequence generator developed in this chapter was first implemented in Simulink, then converted to SynplifyDSP, and finally synthesized into VHDL for hardware validation in a Xilinx Virtex 4 LX60 FPGA[120]. All sequence values proved to be "bit-true" to the SynplifyDSP simulation models, using the hardware resources captured in Table 1. Note that hardware utilization numbers can vary as a result of implementation, clock rates, destination part, and VHDL synthesizer, so are estimated below the lowest level that the VHDL tools report.

TABLE 1. Prototype digital chaotic circuit hardware utilization.

Component	**XtremeDSP Slices**	**Registers**	**LUTs**	**BRAMs**
Chaotic Ring Generators	0	1728	2608	16
Masking Ring Generators	0	1856	2640	16
Box-Muller Transformation	5	423	1143	5
RNS Reduction	0	103	604	1
Control/Glue Logic	1	2013	507	0
Total	6	6123	7502	38

2.4 Analysis of Digital Chaotic Sequences

To validate the conceptual construction and hardware implementation of the chaotic sequence generator, this section discusses a wide range of standard techniques to qualify or quantify the randomness of a random sequence. Unless specifically noted otherwise, it is

assumed throughout this section that the uniformly pseudorandom sequence derived in the preceding sections has already been processed with the uniform to bivariate Gaussian Box-Muller transformation[111], so that the sequence under evaluation is a complex-valued random variable that approximates additive white Gaussian noise.

► **First-order sequence characteristics** include short-run and long-term histograms, visual inspection, and similar qualitative approaches.

► **Statistical randomness tests** should be used to verify that the predicted and measured values match the analytical conditions for a chaotic spreading sequence exhibiting maximal entropy: Gaussian distribution characteristics under all standard tests, correspondance between all cumulants and higher-order statistics, and impulsive autocorrelation/negligible cross-correlation.

► **Time-domain randomness tests** include attempts to fit more complicated time-domain correlation-models to any type of pattern in the sequence. The **a**uto**r**egressive **i**ntegral **m**oving **a**verage (ARIMA) models are effectively a codification of stochastic signal processing.

► **Frequency-domain randomness tests** will primarily apply fast Fourier transform (DTFT) techniques to locate any form of apparent periodicity, color, or statistically significant frequency content.

► **Traditional pseudorandom number generators**[121] (PRNG) will be compared to show the results of the digital chaotic sequence when known reverse engineering techniques are applied.

2.4.1 First-Order Sequence Characteristics

A first-order evaluation of the digital chaotic sequence is worthwhile to both validate some of the common sense predictions for random sequence behavior as well as to introduce the general characteristics that will be quantified more precisely in subsequent sections. To begin, an isolated model of the chaos generator was implemented and initialized using the script included in the Appendix;[21] The digital chaotic sequence was produced at two times the chip rate of 10 MHz and run for 400 ms in order to collect 8 million samples, which is $1.6 \cdot 10^{-75}\%$ of the total sequence length. In a most extreme case, assume 10 billion users each declared ownership of a distinct portion of the sequence (defined by a private key/initial condition) for the next 100 years, and the sequence is generated at rates up to 1 GHz, then collectively only $6.3 \cdot 10^{-54}\%$ of the sequence would be used. The collection of sequence depictions in Figure 30 include (a) a Monte-carlo scatterplot for the first 10,000 uniform sequence outputs, (b) a time

[21] A random seed to Matlab's random number generation was selected via *rand('state',42);*.

domain snapshot of the Gaussian output, and (c) a 10,000-bin histogram of the 8-million point Gaussian sequence compared to the expected envelope (red) of a standard normal distribution.

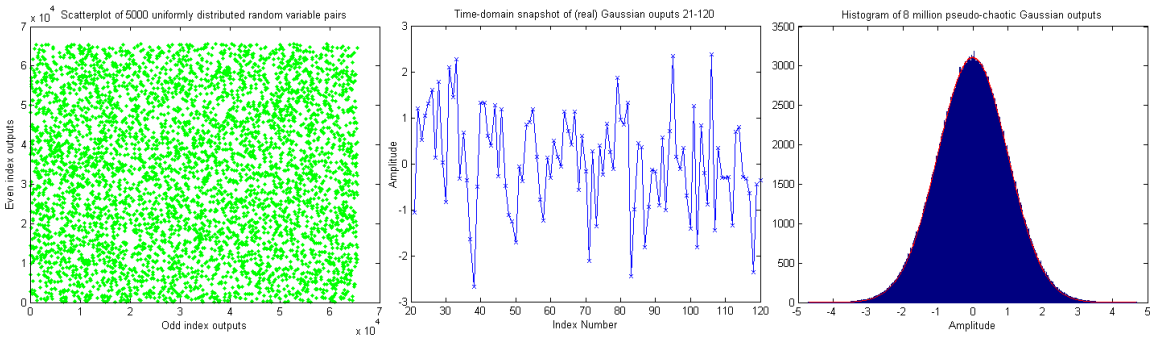

FIGURE 30. First-order characteristics of the digital chaotic sequence.

The chaotic sequence clearly meets the common sense tests for randomness; the subsequent sections will introduce specific mathematical methods for evaluating randomness of the chaotic sequence, applying those techniques to quantify the fit of the arbitrary 400 ms collection. This run has been repeated using arbitrary initial conditions, resulting in statistically identical results.

2.4.2 Statistical Tests of Randomness

The first set of tests for whether a sequence is sufficiently random is whether the short-run and long-term sequence characteristics fit the statistical models. With a sample size of up to 4 million complex-valued elements, the statistical characteristics should converge satisfactorily to the analytical models. A number of emperical tests[122] for randomness (t-statistics, coin-flip models, run lengths, etc.) have been successfully performed in addition to the analytical models. This section concentrates on analytical models since any two random processes are identical when the moment generating functions (equivalently the moments or cumulants) are identical for all measurable content[123]. Specifically, this section focuses on cumulants, which are convenient engineering tests for detecting deviations from maximal signal entropy.

The ideal coherent chaotic waveform is featureless, which is equivalent to maximal entropy since any statistically significant quantifiable waveform feature represents a reduction in signal entropy. The attributes that are exploited in waveform feature can be any type of time-domain, frequency-domain, or statistical artifacts that provide partial information into the signal's nature or content. As a simple example, a QPSK-modulated signal can be quadrupled to obtain reliable metrics on the symbol period or bandwidth. Other methods of detecting features and classifying unknown signals rely on stochastic artifacts that are more difficult to mask –

delayed correlations to obtain symbol timing, matrix pencil algorithms, and general higher-order statistics. Of particular interest are the kurtosis and skewness of a sample sequence, which give insight into the type of analog/digital filtering used, the modulation type, and the statistical distribution of the signal values.

To be an ideal maximal entropy waveform, any chaotic waveform must have an excess kurtosis of 0 (the same kurtosis of AWGN) and a skewness of 0 in addition to the more common metrics of zero-mean and homoskedastic normalized unit-variance. The skewness of a sequence is measured as the third central moment divided by the cube of the standard deviation. Stated another way, the skewness is the third cumulant divided by the $\frac{3}{2}$ power of the second cumulant. Similarly, the excess kurtosis of a sequence is the amount that a sequences' kurtosis deviates from that of an ideal normal distribution (kurtosis of 3), measured as the fourth central moment divided by the fourth power of the standard deviation.

Let μ_x represent the mean and σ_x represent the standard deviation of the sequence x . Let $\mu_{x,k}$ represent the k^{th} central moment of a sequence for any $k \geq 2$, which is calculated as

$$\mu_{x,k} = E[(x - \mu_x)^k]$$

A common application of cumulant calculations is the sequence variance, which is the second cumulant, $\mu_{x,2}$, and is equal to $\sigma_x^2 - \mu_x^2 = \sigma_x^2$ for zero mean sequences. The skewness and kurtosis are the next higher-order statistics considered to detect signal features and are defined as

$$\text{Skewness:} \quad \frac{\mu_{x,3}}{(\mu_{x,2})^{\frac{3}{2}}} = \frac{\mu_{x,3}}{\sigma_x^3} \qquad \text{Excess Kurtosis:} \quad \frac{\mu_{x,4}}{(\mu_{x,2})^2} - 3 = \frac{\mu_{x,4}}{\sigma_x^4} - 3$$

The skewness of a statistical distribution quantifies the asymmetry of the distributions' tails. One way to visualize this is by considering a center of mass calculation, where skewness represents the amount that the mass center deviates from the distance center or mean. Two arbitrary probability density functions are shown in Figure 31, compared to the Gaussian distribution, which has an ideal skewness of 0.

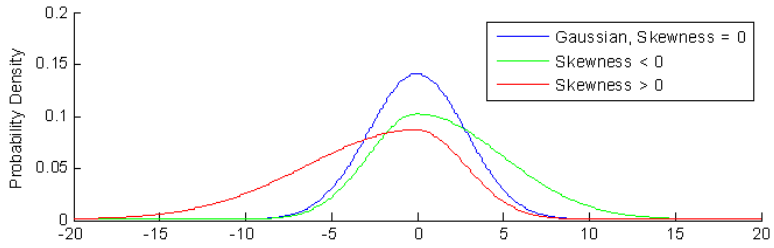

FIGURE 31. Skewness comparison of probability distributions.

In general, the features of the chaotic spreading sequence ensure that the skewness, and more generally all odd-order cumulants, will always be zero. As proof, consider the general case of an odd-order cumulant, $\mu_{x,k}$, where $k \in \{1, 3, 5, ...\}$. The symmetry of the Gaussian distribution ensures that the expected value of a sample raised to the k^{th} power has a magnitude that depends only on the absolute value of the sample and a sign that is uniformly chosen from $\{-1, 1\}$. The limiting value of a symmetric summation of such values will necessarily cancel each other out, making all odd-order cumulants converge to zero.

The more mathematically precise language associated with the kurtosis measure, which is effectively an indicator of peakedness in a statistical distribution, breaks down the result by the sign of the excess kurtosis.

$$\text{Excess Kurtosis } (K): \quad \begin{cases} K < 0 & \text{Platykurtotic} \\ K = 0 & \text{Mesokurtotic} \\ K > 0 & \text{Leptokurtotic} \end{cases}$$

A graphical display of platykurtotic, mesokurtotic, and leptokurtotic probability distributions is shown in Figure 32.

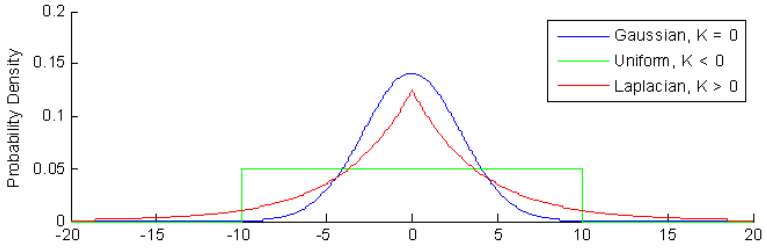

FIGURE 33. Kurtosis comparison of probability distributions.

There are rare exceptions where non-Gaussian distributions can be mesokurtotic, yet those distributions will have other characteristics that can be exploited to distinguish them from AWGN when applied as a waveform. Common "distributions" used in waveform design include raised-cosine (platykurtotic), Laplacian (leptokurtotic), and uniform (platykurtotic). A summary of the higher-order statistical behavior for common distributions is shown in Table 2.

TABLE 2. Stochastic features of various distributions.

Statistical Distribution	**Skewness**	**Excess Kurtosis**
Bernoulli	$\frac{1-2p}{\sqrt{p(1-p)}}$	$\frac{1}{1-p}+\frac{1}{p}-6$
Exponential	2	6
Gaussian	0	0
Laplacian	0	3
Rayleigh	$(\pi-3)\sqrt{\frac{\pi}{2\left(2-\frac{\pi}{2}\right)^3}}$	$\frac{6\pi(4-\pi)-16}{(\pi-4)^2}$
Uniform	0	$-\frac{6}{5}$

It is relatively simple to construct a recursive relationship for all cumulant values of an ideal Gaussian distribution, providing a basis for comparison to the digital chaotic sequence. In general, the k^{th}-order cumulant is calculated as the ratio of the expected value of x^k (i.e. the k^{th} central moment) to the k^{th} power of the standard deviation.

$$\text{k}^{\text{th}}\text{-order cumulant:} \quad \frac{\mu_{x,k}}{(\mu_{x,2})^{\frac{k}{2}}}$$

Calculation of the higher order cumulants follow from an inductive approach using integration by parts. The probability density function for the Normal distribution is

$$pdf(x) = f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The calculation of the k^{th} central moment[22] is computed as the expected value of x^k, or

$$E[x^k] = \mu_{x,k} = \int_{-\infty}^{\infty} x^k\, f(x)\, dx = \int_{-\infty}^{\infty} \frac{x^k}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}\, dx$$

By applying the standard integration by parts rule

$$\int_a^b u\, dv = uv\big|_a^b - \int_a^b v\, du$$

and choosing

$$u = x\, f(x) \qquad \Longrightarrow \qquad du = \left[f(x) - \frac{x^2}{\sigma^2} f(x)\right] dx$$

$$dv = x^{k-1}\, dx \qquad \Longrightarrow \qquad v = \frac{x^k}{k}$$

Solving the resulting equation gives an recursive formula for the k^{th} even central moment, for all $k \geq 0$. All odd-order cumulants are equal to 0, both by direct computation and by simple symmetry arguments, while even order cumulants are necessarily positive. The two necessary initial conditions for the recursive relationship are $\mu_{x,0} = 1$ and $\mu_{x,1} = 0$, calculated

[22] "Central" moment refers to a zero-mean distribution, or more precisely $E[(x-\mu_x)^k]$, which are identical when $\mu_x = 0$.

as

$$\mu_{x,0} = \int_{-\infty}^{\infty} x^0 \, f(x) \, dx = \int_{-\infty}^{\infty} f(x) \, dx = 1 \qquad \mu_{x,1} = \int_{-\infty}^{\infty} x^1 \, f(x) \, dx = \mu_x = 0$$

As a result, all odd-order cumulants are equal to zero for the Normal distribution and even-order cumulants are constant multiples of the k^{th} power of σ.

$$\mu_{x,k} = \frac{k!}{\left(\frac{k}{2}\right)! \, 2^{\frac{k}{2}}} \sigma_x^k \qquad k \geq 2$$

Further, practical receive precision in a communication system is limited to approximately 16 bits, making computation and comparison of any cumulants higher than 8^{th} order prone more to numerical precision errors than to statistically significant decisions on receive data. A chaotic sequence that is mapped onto a quadrature Gaussian sequence therefore need only be shown to match the lower-order statistics before being considered chaotic within measurable limits.

One thousand random sequences of two million samples each was generated by a simulation of the digital chaotic circuit[23] and analyzed to determine the statistical behavior as measured by the cumulants. A comparison of the digitally generated chaotic sequence to Matlab's internal Gaussian random number generator[124] and also to the ideal cumulant values is shown in Table 3; the chaotic sequence shows strong agreement in all cumulants up to eighth order, making it virtually indistinguishable from additive white Gaussian noise.

TABLE 3. Cumulant comparison between standard normal distribution and digital chaotic sequence

Cumulant	**Standard Normal**	**Chaotic Sequence Mean**	**Chaotic Sequence Std Dev**	**Matlab PRNG Mean**	**Matlab PRNG Std Dev**
$1^{\text{st}}: \frac{\mu_x}{\sigma_x}$	0	0.000910	0.000464	0.000023	0.001003
$2^{\text{nd}}: \frac{\mu_{x,2}}{\sigma_x^2}$	1	1.000024	0.000980	1.000008	0.001520
$3^{\text{rd}}: \frac{\mu_{x,3}}{\sigma_x^3}$	0	0.000056	0.001734	0.000067	0.002419
$4^{\text{th}}: \frac{\mu_{x,4}}{\sigma_x^4}$	3	3.000869	0.003433	2.999878	0.004962
$5^{\text{th}}: \frac{\mu_{x,5}}{\sigma_x^5}$	0	0.000443	0.018844	0.000788	0.026746
$6^{\text{th}}: \frac{\mu_{x,6}}{\sigma_x^6}$	15	15.01862	0.054391	14.99873	0.079504
$7^{\text{th}}: \frac{\mu_{x,7}}{\sigma_x^7}$	0	0.001806	0.243085	0.014073	0.352941
$8^{\text{th}}: \frac{\mu_{x,8}}{\sigma_x^8}$	105	105.3106	0.872040	104.9866	1.304348

[23] An exemplary Matlab script is included in the appendix; samples are quantized based on the measured NLP performance to emulate chaotic sequence generation.

2.4.3 Time-Domain Tests of Randomness

Various time-domain models exist for detecting autocorrelations in a sequence of values and using those relationships to predict future values. One of the more common techniques used is the autoregressive integral moving average (ARIMA) models[125] that try to fit the data with some past relationship of itself. The autoregressive portion of ARIMA relies on autocorrelations of the sequence with itself; these models are common in trying to detect seasonality or even patterns in stock prices. The integral, or integrated, portion of ARIMA extends the autocorrelation search to an analysis of the changes in the sequence at each index rather than the actual values. The moving average portion constructs linear combinations of past "shocks" into a meaningful fit of data to predict future trends based on recent system shocks.

A key assumption of the Box/Jenkins ARIMA approach[126] is that a sequence be stationary; oftentimes, the sequence is first differenced to take away a trend, and then the parametric model fit to a stationary sequence is integrated back to form the predictive model for the system. In trying to claim that a sequence is suitably random, the sequence should be stationary, which implies it is "characterized by a kind of statistical equilibrium around a constant mean level as well as a constant dispersion around that mean level[125]." The digital chaotic sequence is notionally a quadrature pair of standard Normal random variables with zero mean and unit variance. A further assumption of strictly stationary time series will in addition have a constant autocovariance structure. The expected result of the chaotic sequence is an impulsive autocorrelation, which is consistent with the definition of strict stationarity. An additional condition for a suitably random process is that the cross-correlation with any other physical process be negligible; the list of possible processes is infinite, so will only be addressed briefly. To evaluate these conditions, let $X[k]$ be the discrete-amplitude discrete-time chaotic sequence. The autocovariance of $X[k]$ is defined as

$$\Phi(k) = E[X_t, X_{t-k}] = \sum_{t=1}^{n-k} (X_t - \mu_x)(X_{t-k} - \mu_x)$$

where t is any arbitrarily chosen starting index. The assumption of stationarity requires that there be no dependence on the value of t for autocovariance behavior. The chaotic sequence has already been shown to be sufficiently close to a quadrature pair of standard Normal random variables with zero mean and unit variance. Therefore, the autocovariance and autocorrelation functions are equivalent (normalizing by the number of terms in the summation) and may be simplified to

$$\Phi(k) = \frac{1}{m} \sum_{t=1}^{m} X_t X_{t-k}$$

Should the chaotic sequence have any significant autocorrelations, it would be expected that they occur at multiples of the prime characteristics in the ring generators. Therefore, a significantly long span of samples (2^{16}) was autocorrelated and added to determine if any non-impulsive correlation features exist. The left side of Figure 33 shows the impulsive autocorrelation characteristic over a span of (2^{17}) samples (correlation contour plotted in green with points superimposed in blue), while a closer view (2^{11}) with the central spike included (center) and excised (right) are also included. What appears to be an autocorrelation outlier in the excised figure occurs at delay 547, which is not one of the primes, and has been shown in subsequent tests to be an outlier; frequency-domain tests are used later to validate.

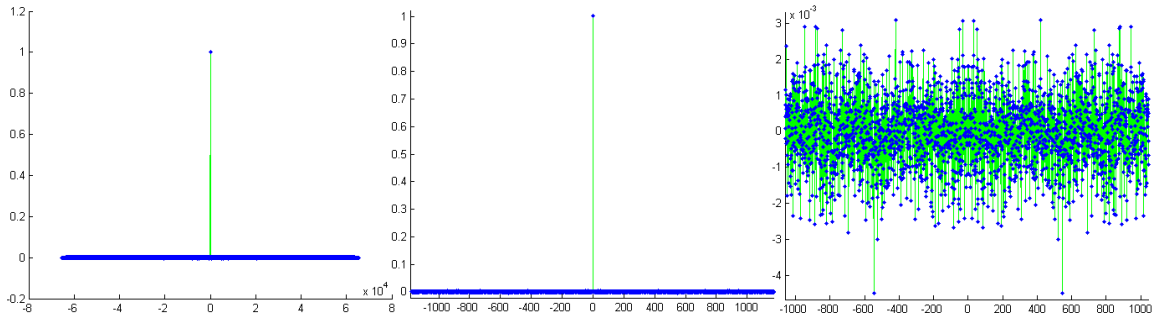

FIGURE 33. Autocorrelation of chaotic sequence over 2^{17} samples (left), 2^{11} samples (center), and with central spike excised (right).

To validate this assumption statistically, a histogram of the autocorrelations was generated, resulting in the expected scaled Bessel function characteristic that comes from the multiplication of two independent Gaussian random variables[112] as depicted in Figure 34.

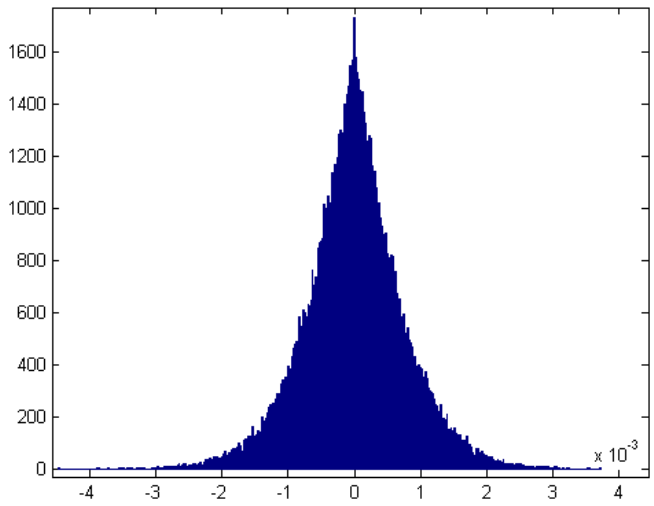

FIGURE 34. Zero-th order Bessel function characteristic of autocorrelation values.

This histogram can be fit by the distribution

$$f(z) = \int_0^\infty \frac{\cos(zt)}{\pi\sqrt{t^2+1}} dt$$

where z is the product of two independent standard normal random variables[127]. Since the autocorrelation function is impulsive, the entire autoregressive portion of the ARIMA models will fail to have statistical significance. Note that the idea of FIR or IIR filtering is mathematically identical to this correlation process (i.e. a matched filter approach), suggesting that a receiver must have the synchronized conjugate of the sequence to obtain a correlation peak or other useful information.

Since the digital chaotic generation process is self-contained and deterministic, there are no induced shocks from which to derive a MA model, leaving only finite difference equations as a possibility. Unlike analog chaotic circuits, clock jitter and many other practical non-idealities do not significantly effect the coherency of synchronized chaotic streams when proper mitigations are employed. Therefore, the next step is to attempt locating a statistically significant pattern in the chaotic sequence using combinations of static finite differences; the expectation is that by taking the first difference of a sequence that approximates a standard normal distribution

$$Y[k] = X[k+1] - X[k]$$

we obtain another random sequence with Gaussian characteristics, but a standard deviation that is $\sqrt{2}$ times as large. Repeating the preceding analyses for comparison to a Gaussian random process holds true, and the resulting histogram of the first difference is shown in Figure 35. Since the standard normal distribution is fundamentally based on the exponential function, which is in turn infinitely differentiable (analytic), it is anticipated that all orders of derivatives (finite difference equations) will similarly result in a lack of time-domain patterns. This process was repeated for differences with relative delays of 1 up to 1000 with statistically identical results, indicating that the digitally generated chaotic sequence has no measurable internal correlations that will reduce the signal entropy when used in a chaotic communication system.

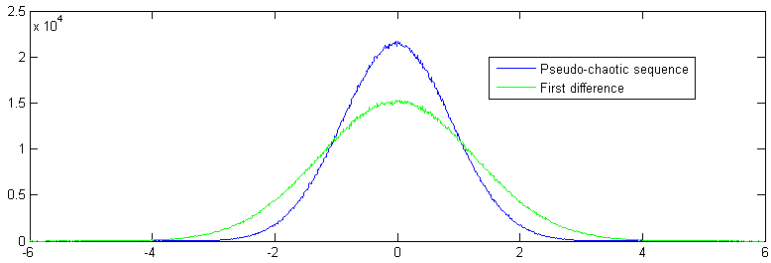

FIGURE 35. Comparative distributions of chaotic sequence and first difference.

2.4.4 Frequency Domain Tests of Randomness

The frequency domain tests for randomness center on the Fourier transform which inherently converts a time-domain sequence into frequency components that may be exploited for signal characteristics/specific frequency content. The traditional definition of the DTFT is given by

$$X(f) = \mathbb{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$$

where the complex exponentials $e^{-j2\pi ft}$ effectively correlate with frequency content in $x(t)$ at each chosen frequency f. The most common usage of the Fourier transform is the discrete time Fourier transform (DTFT), or fast Fourier transform (FFT), that discretizes both the time step dt and the frequency bins df. The application of the Fourier transform to evaluation of the chaotic sequence focuses on the notional "white" noise characteristic of an ideal chaotic signal (similar to Shannon's ideal waveform in a flat channel), where an equal amount of signal energy is contained in any arbitrary portion of the transmission bandwidth. Equivalently, a chaotic signal should have a perfectly flat spectral response without any frequency components that can be exploited for signal detection or move the signal away from Shannon's equal-energy information capacity ideal. A series of 1000 independent DTFT tests were performed on 2 Mpt length random samples of the chaotic sequence, resulting in time-averaged 1 Mpt DTFTs and a clearly logarithmic bin distribution[112]. The time averaged DTFT is shown on the left side and the resulting histogram of the bins (measured in dB) is shown on the right side of Figure 36; in both cases, the expected spectral content for a normalized input is the inverse of the number of points in the DTFT in each frequency bin, or $-20\log_{10}1000000 = -120$dB for the 1Mpt DTFT.

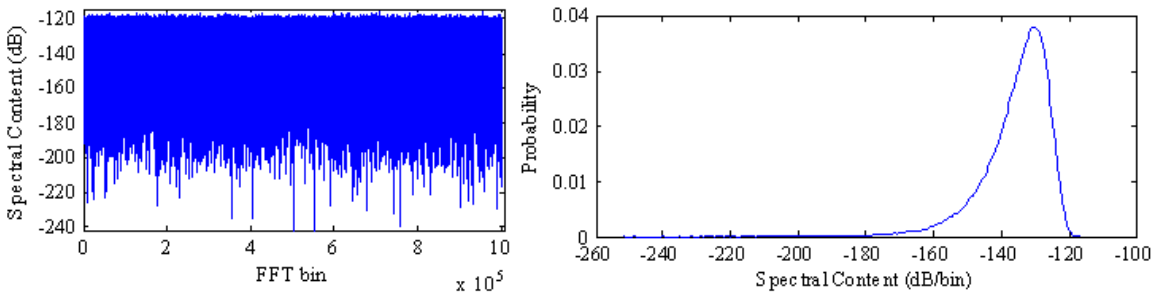

FIGURE 36. Discrete time Fourier transforms of chaotic sequence; 1000 time-averaged 1 Mpt DTFTs (left) with histogram of bin values in dB (right).

The power spectral density of the chaotic sequence as measured by these DTFT measures is exceedingly flat, indicating that there are no significant frequency components or features that represent anything other than maximal signal entropy. Additional frequency domain tests include validation that the Fourier transform of Gaussian random values are again Gaussian distributed (verified as before using cumulants) and that larger DTFT measures (16 Mpt) result in flat power spectral densities.

2.4.5 Comparison of Digital Chaotic Sequence to Traditional PRNGs

Compared to traditional pseudorandom number generators (PRNG), the prototype digital chaotic sequence derived in this dissertation appears to be suitably random and maintain the core properties of a discrete approximation to a chaotic system. The majority of the traditional PRNGs were developed for cryptographic applications, including linear feedback shift registers[103, 104, 102], the DES and AES Rijndael core[128], Mersenne twister algorithms for simulation[124], and others[129] that claim to be cryptographically secure. There are neither any obvious attacks for reverse engineering the digital chaotic sequence, nor is there extensive coverage in this document to show that the sequence is suitable for cryptographic applications without additional components; various tests of randomness covered in the NIST open-domain standards[130, 122] have been applied without locating relevant features or correlations. More importantly, it is believed that any suitably random sequence shaped into a bandlimited AWGN stream will be sufficient for the construction of a chaotic communications spreading signal with maximal entropy. The coherent communications challenge remaining is to efficiently harness multiple copies of the digital chaotic circuit for receiver acquisition and synchronization.

Chapter 3: Practical Chaotic Communications

Given a practically infinite sequence that approximates additive white Gaussian noise, the goal becomes harnessing this sequence to create a robust maximum entropy chaotic communication system. The statistical properties discussed in the previous chapter must be retained throughout the chaotic circuit initialization and control, data modulation, subsequent filtering, data conversion, and RF upconversion. Most of the previous work in chaotic communication systems focused on employing the chaotic circuit to modulate user data; common methods include phase shift keying, pulse amplitude modulation, carrier frequency hopping, and/or combinations thereof. Preferably, the modulation scheme will not change the statistical characteristics of the signal, yet offer compatability with higher capacity modulation schemes like QAM or APSK variants. Whichever type of modulation scheme is used, the receiver must be able to accurately acquire and maintain synchronization of its coherent chaotic sequence to mitigate propagation delay, Doppler effects, and hardware nonidealities. Further, the hardware required to construct the chaotic communication system should be practically implementable: basing the architecture on a software defined radio (SDR) promotes the greatest flexibility and potential for integration into modern communications. Beyond hardware, the system must be able to implement communications protocols and not prohibit exploitation of time, frequency, spatial, or coding diversity. Finally, considerations must be given to application of the chaotic communication system, quantified as frequency re-use characteristics, signal entropy and features, size/weight/power (SWaP), along with any unintended consequences of design choices. Summarizing the ideal characteristics of a practical communication system based on a digital chaotic circuit:

► A **practically infinite chaotic sequence** provides the fundamental code permitting the intended user to receive the information. To approach both Shannon's information capacity ideal noise-like waveform, the transmitted waveform should be indistinguishable from maximal entropy AWGN.

► A **robust chaotic sequence synchronization** method is absolutely required for a chaotic waveform since the sequence has a naturally impulsive autocorrelation. This synchronization scheme must also contend with the traditional nonidealities like frequency offsets, timing offsets, clock jitter, and gain control. The acquisition and synchronization mechanisms for the chaotic waveform are presented in Chapter 4.

► An **efficient data modulation scheme** is required to encode and decode the user information in a coherent fashion that fully utilizes the transmission bandwidth.

► A **flexible communications protocol** is required to support transmission channel variations, including multipath/fading induced dropouts, correcting bit/symbol errors, and meeting simultaneous demands of multiple communications users.

► A **compact spectral response** is preferred so that no stray energy is emitted into frequency bands outside those used for signal reception. These frequency spurs or images reflect reductions in the signal entropy that correspond to less efficient use of the transmitted energy.

► Successful **integration of the RF transmit chain** into the baseband processing is required to mitigate the effects of D/A conversion, frequency upconversion from IF to RF, and transmission through the antenna.

► An **optimized receiver architecture** that adapts to signal dynamics and can efficiently convert the spread wavefom to meaningful information. The basic structures for frequency, phase, and time tracking are derived as generalizations of direct sequence spread spectrum receiver technology.

One of the key considerations in constructing communication systems is how the intended signal will be discriminated at the receiver. At the most fundamental level, there are only a few practical mechanisms by which to encode information in an electrical signal, each of which having advantages and disadvantages. Moreover, there is a desire at the receiver to discriminate between the transmitted signal and an unknown selection of background noise and interfering signals. These discrimination methods[131] may be categorized into time diversity, frequency diversity, spatial diversity, or coding diversity. Each has advantages from implementation, spectral re-use, and data throughput viewpoints.

► **Time Diversity:** The entire concept of packetized communications relies on finite bursts of RF energy that have distinct start and stop times, permitting an ordered sequence of bursts from one or more users in different "timeslots." A simple example of a system that depends on time diversity is TDMA telephony.

► **Frequency Diversity:** Many communications depend on channelization where two or more parties are transmitting simultaneously on different RF carrier frequencies. Using a frequency tuner and filtering, receivers are capable of selecting their desired channel. Two simple examples of systems that depend on frequency diversity are AM or FM radio broadcasts.

► **Spatial Diversity:** Spatial diversity is harder to control than time or frequency diversity and relies on either directional or spatial characteristics for optimized reception. Any

system that uses phased antenna arrays, directional antennas, or power control algorithms based on distance/angle exploit spatial diversity.

► **Coding Diversity:** Coding diversity is the most ethereal; by use of orthogonal codes, multiple users can simultaneously reuse the same frequency spectrum, separating their intended signal from the others by mathematical correlations and coding gains. A common example of a system that depends on coding diversity is CDMA telephony.

The primary mechanisms used for the prototype coherent chaotic communication system are spread spectrum CPSK modulation that encodes data as quadrature phases, relying on the inherent code diversity at the receiver to perform acquisition and synchronization. That combination offers the greatest potential for harnessing a chaotic circuit to fully utilize the channel capacity. This chapter discusses the overall structure, analytical development, practical implementation, and comparison of simulated and measured performance for what is believed to be the world's first practical coherent chaotic communication system; the design and analysis of the communication system was developed under the guidance of Harris advisor David Chester,[24] while the hardware implementation of the design represents a collective R&D effort of the Harris team. Proprietary details of the design are intentionally excluded, with this chapter serving as a firm foundation for the broader development of chaotic communication system presented in the remainder of this document. Specific advances covering the digital chaotic circuit synchronization, a novel generalized spread spectrum acquisition approach, and core modifications to spread spectrum receiver design are covered in the next chapter. Subsequent chapters focus on the extensions of the core coherent chaotic communications technology to construct practical chaotic waveform variants, including multiple access technology, CAZAC ranging waveforms, generalized chaotic modulation of arbitrary data constellations, generalized chaotic signaling bases, and mitigation techniques for transmission effects.

3.1 Prototype Chaotic Commmunication System Frequency Plan

The starting point for any communications system is the choice of waveform and the frequency plan. Among the considerations, the spreading sequence generation rate (transmitted signal bandwidth) was constructed at 10 MHz so that the baseband processing may operate on multiple samples per chaotic sequence chip; this ensures that the transmitted signal may be bandlimited and also not adversely impacted by the finite sampling rate of the D/A converter

[24]David Chester is a Sr. Scientist at the Harris Corporation in Melbourne, FL; he earned his PhD in Computer Engineering at the University of Cincinatti and has over 30 published papers in signal processing and communications to go with 12 patents. He is an adjunct faculty member at the University of Central Florida and Florida Institute of Technology.

(240 MHz). This signal is then passed through a halfband interpolating filter, used to modulate the data stream (does not change spectral characteristics), passed through a rate change filter and then presented to the upconversion stage. The digital upconversion takes the baseband signal, translates it to a standard IF of 70 MHz and then inserts to an interpolating DAC for upconversion to the 2.4 GHz ISM band. The digital transmit portion of the frequency plan is summarized graphically in Figure 37.

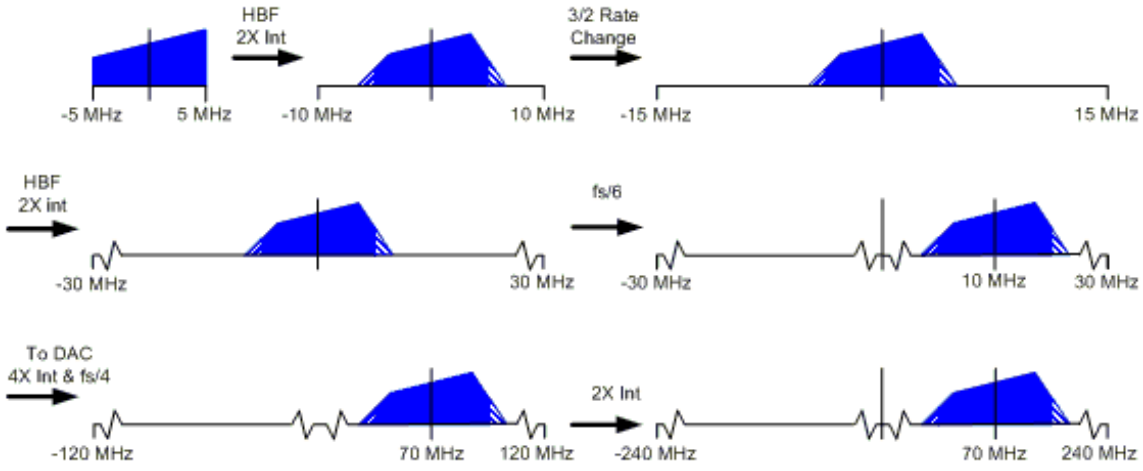

FIGURE 37. Digital portion of transmit frequency plan.

The interpolating D/A conversion process has a sinc characteristic that causes rolloff within the IF bandwidth and must be compensated in the design of the preceding filtering stages. Two simulations were completed to determine the impact caused by the D/A, with one at a D/A rate of 240 MHz, and the second at a D/A rate of 480 MHz. The worst-case rolloff in the first case is 0.38 dB within the 10 MHz signal bandwidth, while the worst-case rolloff in the second case is only 0.09 dB; neither of these rolloffs (shown in Figure 38) are expected to significantly impact the waveform characteristics.

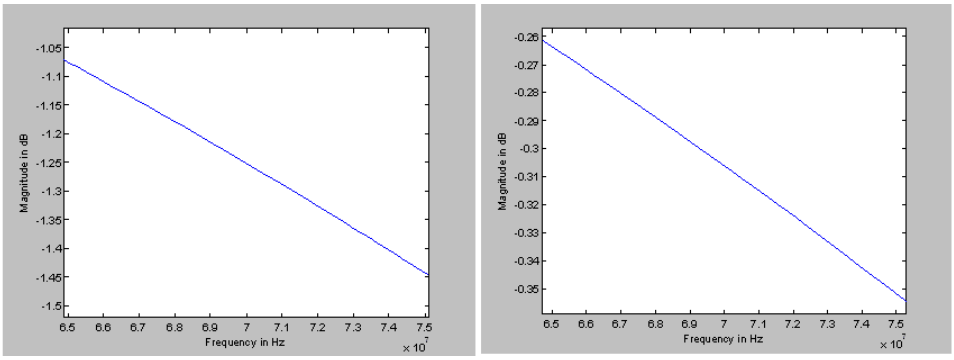

FIGURE 38. Rolloff characteristics of interpolating DAC at $f_s = 240$ MHz (left) and $f_s = 480$ MHz (right).

The RF upconversion is inverted by the receiver block downconverter returning the S-band signal to a 70 MHz IF, which is then subsampled and $\frac{f_s}{4}$ digitally downconverted to a pair of signals (signal and a negative frequency image) at 10 and 30 MHz. The signal at 10 MHz is isolated by a decimating halfband filter and $\frac{f_s}{4}$ downconverted to baseband; the baseband signal has a 10 MHz bandwidth and a sample rate of 40 MHz. These signal manipulations are depicted graphically in Figure 39.

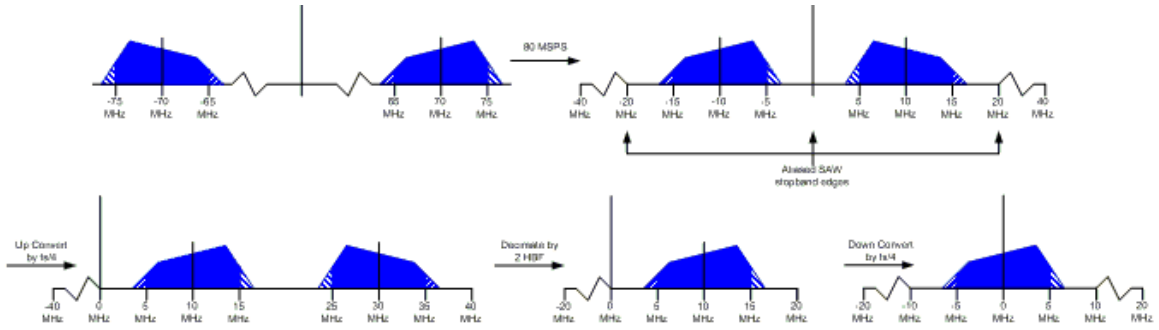

FIGURE 39. Digital portion of receive frequency plan.

Throughout the description of the frequency plan, a primary consideration for efficient implementation drives the seemingly unneeded center frequency or bandwidth manipulations. In all cases, multirate filter techniques for efficient signal manipulation are used. One specific example of efficient filtering is the halfband interpolation filters at the output of the chaotic sequence generator; by using a symmetric halfband filter, half of the coefficients may be discarded since they will be identically zero and the other half may be pairwise combined using pre-additions. Likewise, the upconversions and downconversions on both side of the communications link take advantage of sparse coefficient multiplications that may be implementing physically with switching topologies. One specific example is the Weaver modulator[132] that implements digital downconversion using switches and sign inversions based on a mathematical premise comparable to $\mathbb{Z}_4$ rings. These filtering techniques affect the selection of filter coefficients, while a second technique converts static coefficient multiplication to a specialized structure of shift-adds; this canonic signed digit (CSD) filter implementation is similar to the modified Booth multiplier algorithm[91], and enables implementation of filters with zero hardware multipliers. More discussion as to the implementation techniques is included in subsequent sections.

3.2 Prototype Chaotic Commmunications Transmitter

Widespread recognition that a theoretical chaotic communication system can be constructed from a chaotic circuit began in the early 1990s, and since then, various demonstrations

of a chaotic transmitter have be implemented. The fundamental understanding of the data modulation process, predicted effects and performance in different transmission channel conditions, and a limited amount of implementation criteria have been established. The chaotic phase shift keying modulation shows the greatest practical applicability to coherent communication system design, with theoretical $\frac{E_b}{N_0}$ performance approaching that of traditional PSK modulations and consists of a relatively simple modulation mechanism. This section focuses on the design and practical implementation of a chaotic phase shift keying transmitter that will form the first stage of the protoype coherent chaotic communication system. Since creating a chaotic waveform has been previously achieved[133, 68, 134], the emphasis is placed on creating efficient signal processing techniques that ensure the discrete-amplitude discrete-time chaotic sequence retains its maximal entropy characteristics once modulated and emitted. An evaluation of the analytical/simulated output waveform and comparison to measured hardware results is provided.

Practical implementation of the chaotic phase shift keying (CPSK) waveform requires signal processing techniques that compensate for timing uncertainty, fixed point arithmetic, and secondary effects of all operations. The end goal is to modulate user data in a manner that can be demodulated intelligibly at the receiver, yet be indistinguishable from bandlimited AWGN in the transmission channel. The burden of the signal acquisition and synchronization is placed on the receiver, which must contend not only with nonidealities in its own hardware, but with the phase, frequency, and timing drifts that occur in the transmission channel. The discussion in this section focuses on the exemplary CPSK transmitter architecture shown in Figure 40.

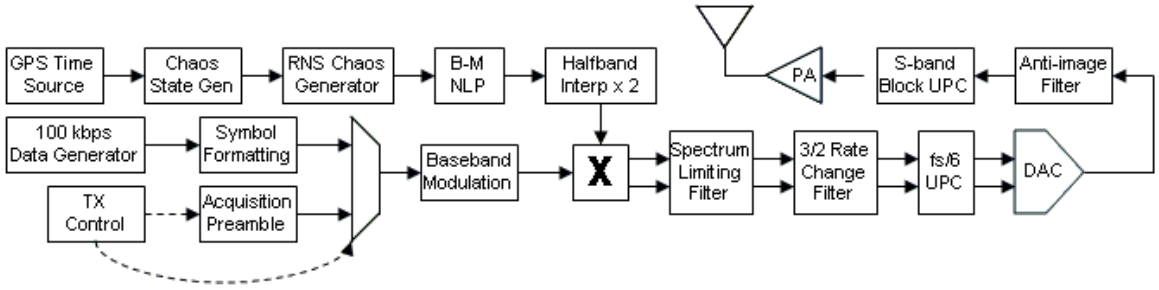

FIGURE 40. Block diagram of CPSK transmitter architecture.

The overall architecture implements a 100 kbps quadrature CPSK waveform that references internal timing to an external GPS module, constructing a chaotic signal per the previous section. Walking through the signal flow, the data generator produces a constant stream of user data at a 100 kbps rate that is formatted into quadrature phase shift keying (QPSK) symbols at a 50 ksps rate. These data symbols are enabled/disabled by a transmit controller that

prepends an acquisition preamble to the beginning of the transmission; higher level protocol functionality like forward error correction (FEC), symbol interleaving, or formal packet structures are omitted in favor of physical layer processing. The stream of data symbols/preamble are phase modulated by the digital chaotic sequence at a 20 MHz rate over a 10 MHz spread bandwidth, resulting in a symbol spreading ratio of $\frac{10 \text{ MHz}}{50 \text{ kHz}} = 200$; the digital chaotic sequence generation chain is as described in Chapter 2, yet is modified to accept external chaotic state parameters and GPS time reference. After data modulation, the remainder of the transmit chain provides filtering and upconversion of the baseband signal to a bandlimited 10 MHz signal centered at an IF of 70 MHz and an RF frequency near 2.4 GHz.

3.2.1 Data Source and Symbol Formatting

The data source and symbol formatting blocks provide a QPSK formatted symbol that can be directly phase modulated (complex multiplication) by the chaotic spreading sequence. The communications performance should never be affected by the data content, so the data generator is equivalent to arbitrary data produced by an unassociated PRNG; all higher-level protocol functions have occured prior to the symbol formatter. This data stream is triggered based on a pulse from the chaotic sequence generator (aligning the data symbol edges with the chaotic sequence time index) and enabled after a preamble of 200 "00" symbols; this preamble provides a known data sequence for the coherent receiver to lock onto and instantiate its phase/frequency/time tracking loops. At the conclusion of the preamble, a pair of disambiguity symbols are transmitted to assist the receiver in receiving a selectable frequency inverted/non-inverted signal. The symbols are formatted using a traditional Gray code as shown in Figure 41, ensuring that most symbol errors result in the error of only one bit.

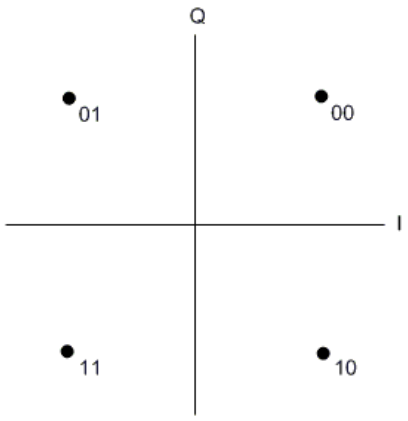

FIGURE 41. Gray-coded data symbol.

3.2.2 Chaotic Sequence Generation

The chaotic sequence generator, which includes the Box-Muller transformation NLP, provides a steady stream of quadrature standard normal random variables for phase modulation of the QPSK-formatted data symbols. The additional control that is needed for practical implementation of the chaotic sequence generator in a transmitter is the ability to externally command the "chaos state" to any arbitrary value as well as enabling/disabling the chaotic sequence generator to burst on precise time intervals synchronized to an external GPS time reference. To set the state of the chaotic sequence generator, it is assumed that both the transmitter and the receiver share an initial condition (key) that is loaded during initialization of the sequence generator; the state implied by this key evolves over time, so that the state of the chaotic sequence generator at any prospective transmission burst is a function of the initial key and the time elapsed since a chosen GPS epoch. Without loss of generality, this epoch was calibrated with that defined by the external GPS time reference, and all future states were indexed relative to that initial time. The difference of this time is recoded[25] from a GPS-centric UTC format of weeks, milliseconds, and sub-ms to an integer number of 20 MHz cycles that have elapsed since the defined time epoch. Using the inherent periodicity of the underlying residue number system, this integer representing the time elapsed is then reduced modulo each of the primes in the RNS-based chaotic sequence generator to determine the residual time elapsed relative to each prime characteristic.

To calculate the residual cycle component elapsed, the input time elapsed is recursively reduced by $2^m p_i$ whenever the difference is positive. Note that m ranges from 1 to 40 in a decreasing fashion since $p_i \geq 2^8 \, \forall i$ and the input is intentionally constrained to less than 2^{48}. This modular reduction approach can be improved slightly for specific primes, but is better left as a generalized calculator. Calculation of the equivalent prime residues for time elapse requires 41 cycles per each of 32 primes, for a total of 1312 cycles at 20 MHz, or 65.6 μs. That processing latency combined with the command/receipt time of the GPS time message sets the lower bound on the discretization of the transmit burst timing unless scheduled bursts are implemented. The final modification to the chaotic sequence generator from that shown previously is the creation of a 50 kHz symbol clock derived from and synchronized with the timing of the chaotic sequence. The architecture for this clock is exactly the same as the ring generator used in the chaotic masking sequence except that a binary output is generated from a comparison of the internal ring value.

[25]The time epoch of the GPS reference requires approximately 60 binary digits to represent accurately, while the recoded time implementation retains only 48 bits, reducing the effective epoch period to only 162 days. This reduction was intentionally created to validate the arithmetic using floating point "double" data structures.

3.2.3 Chaotic Transmitter Filtering

Two of the key components in the chaotic communications transmitter are a pair of odd-order halfband interpolation filters constructed using canonic signed digit coefficient multiplications. These filters are efficient in multiple dimensions:

► Even-order coefficients, with the exception of the center tap are all zeros. The center tap is $\frac{1}{2}$, which may be implemented with a binary shift.

► Alternating output samples, which occur at twice the rate of the input, switch between the sum of even terms (a single tap) and sum of odd terms (multiple coefficients), providing efficient multirate filter phase selection.

► Odd coefficients are symmetric around the center tap, permitting reduction in 50% of coefficient multiplications via pre-additions of symmetric taps.

► Canonic signed digit (CSD) coefficient multiplications trade design effort for optimized hardware, hardcoding the traditional modified Booth[91] multiplier recoding processes[92] for the static filter coefficient multiplications.

The design of these filters can be simplified by recognizing that the passband and stopband are at equal distances from the ideal brickwall cutoff frequency, leading to symmetry in both the time domain and frequency domain; the desire to have extremely high rejection of out-of-band frequency content does lead to relatively high-order filters.

3.2.3.1 Chaotic Sequence Halfband

The spectrum of the 10 MHz quadrature Gaussian digital chaotic stream extends directly to the $(-\frac{f_s}{2}, \frac{f_s}{2}) = (-5, 5)$ MHz boundaries, which would cause a small amount of aliasing during the phase-shift keying process if not limited. Alternatively, the chaotic sequence benefits from an interpolation filter that eliminates the highest frequency content before mixing with the data sream. The desired cutoff for this filter is approximately 80 dB, which is the upper limit of the output D/A dynamic range; the resulting filter design is a 46-tap halfband using the standard Parks-McClellan filter synthesis algorithm[135]:

```
bh = firpm(46 , [0 0.4 0.6 1] , [1 1 0 0] )
bh = round( (2^20) * bh ) / (2^20);
bh(2:2:46) = 0;
bh(24) = 0.5;
```

with normalized frequency response shown in Figure 42.

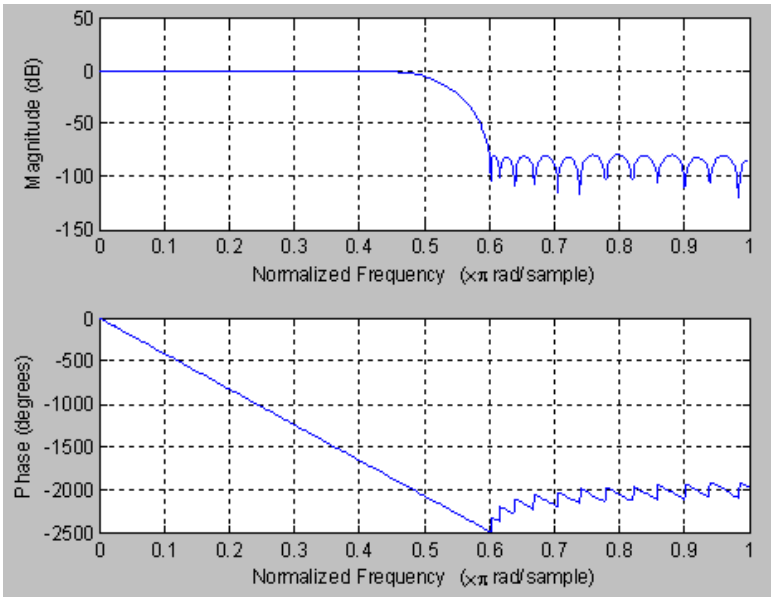

FIGURE 42. Frequency response for chaotic sequence halfband filter.

The implementation of this filter takes advantage of the ability to discard even-order coefficients other than the middle tap, trivially implementing a two-phase interpolate-by-two filter when the output is alternated and sampled at 20 MHz. Further, the remaining coefficients are symmetric about the center tap, leading to the ability to pre-add values before coefficient multiplication.[26] To show this effect more clearly, consider the comparison of unaltered filter coefficients and 20-bit quantized replacements shown in Table 4.

[26]Prior to implementation, a constant gain of 2 is applied to all coefficients, eliminating the multiplication by $\frac{1}{2}$ on the center tap.

TABLE 4. Chaotic sequence interpolate-by-two halfband filter coefficients.

Coefficient	Floating-Point Value	Fixed-Point Approximation
C(0), C(46)	-0.000197700763991	-0.000197410583496
C(1), C(45)	-0.000000213979481	0
C(2), C(44)	0.000576433378955	0.000576019287109
C(3), C(43)	0.000000422951387	0
C(4), C(42)	-0.001351457211000	-0.001351356506348
C(5), C(41)	-0.000001059892400	0
C(6), C(40)	0.002728362087103	0.002728462219238
C(7), C(39)	0.000001702570652	0
C(8), C(38)	-0.004987050923184	-0.004986763000488
C(9), C(37)	-0.000002802870046	0
C(10), C(36)	0.008498242686419	0.008498191833496
C(11), C(35)	0.000003938522016	0
C(12), C(34)	-0.013787094923738	-0.013787269592285
C(13), C(33)	-0.000005387432326	0
C(14), C(32)	0.021711842710723	0.021712303161621
C(15), C(31)	0.000006688502465	0
C(16), C(30)	-0.033978732225803	-0.033978462219238
C(17), C(29)	-0.000008032022203	0
C(18), C(28)	0.054943907772943	0.054944038391113
C(19), C(27)	0.000008893318189	0
C(20), C(26)	-0.100656701419896	-0.100656509399414
C(21), C(25)	-0.000009560236719	0
C(22), C(24)	0.316457153020626	0.316456794738770
C(23)	0.500009741604978	0.5

The coefficient multiplications may be advantageously implemented using CSD representation rather than hardware multipliers. Take for example the first filter coefficient $C(0) = -0.000197410583496$,

$$C(0) = -0.000197410583496 \approx -2^{-12} + 2^{-14} - 2^{-16} = -0.00019836$$

which is accurate to approximately 20 bits. As a result, the pair of identical static multiplications for $C(0)$ and $C(46)$ may be implemented using the pre-add, three binary shifts, and a three input adder as shown in Figure 43.

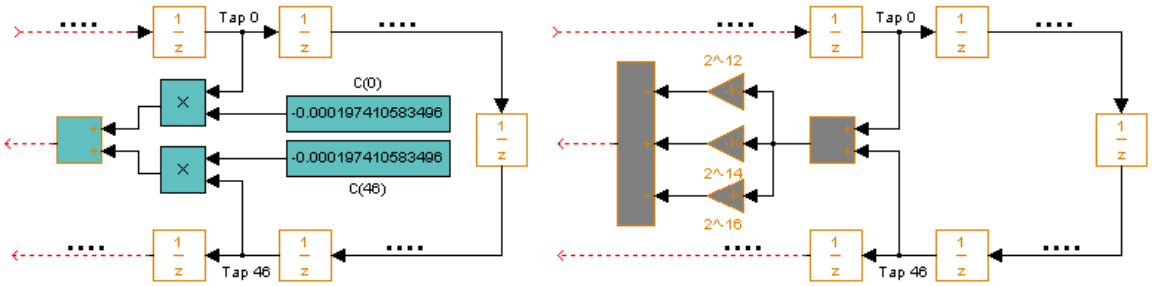

FIGURE 43. Exemplary CSD hardware reduction in filter coefficient multiplication.

The folded filter topology for the overall halfband filter is shown in Figure 44.

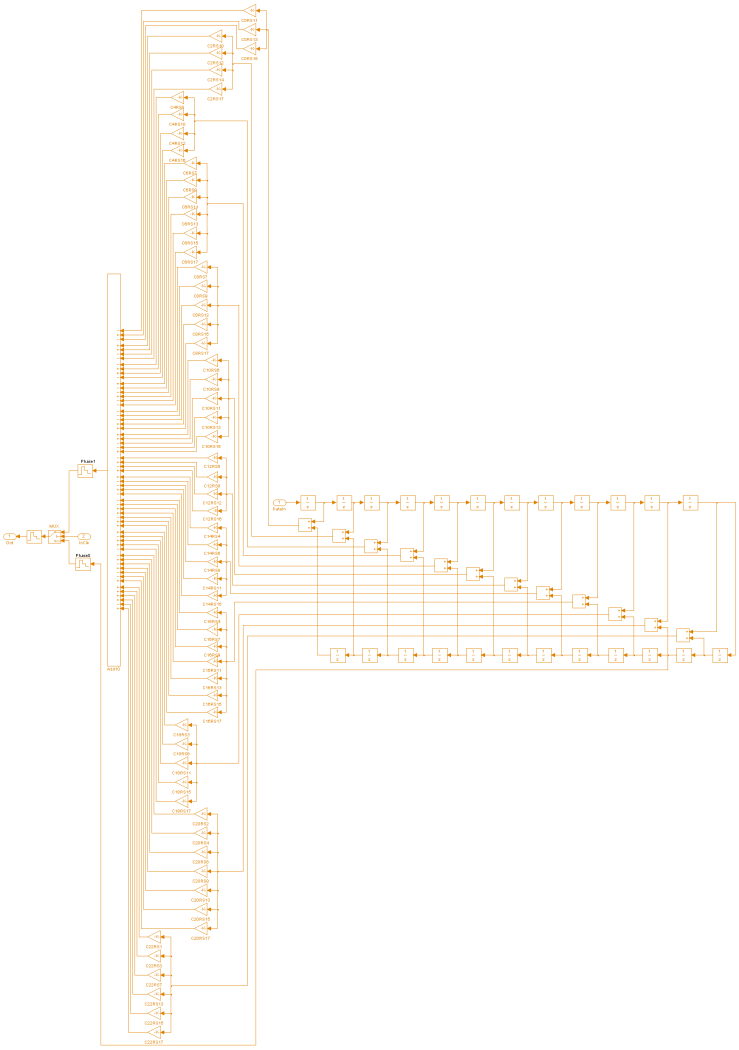

FIGURE 44. Filter topology for folded halfband CSD implementation with pre-adds.

This halfband filter implements the desired filter with an equivalent quantization of 20 bits using 61 shift-additions in lieu of 12 hardware multipliers, assuming pre-adds are using in both scenarios; based on gate count, each multiplier requires the equivalent of 10-11 shift-additions, making CSD implementation a 50% hardware savings. Moreover, VHDL synthesis and mapping tools will automatically winnow down the required adder input widths based on the addition output precision.

3.2.3.2 TX Spectrum Lowpass Filter

Since the QPSK symbol rate is 50 ksps and the chaos sample rate is 20 Msps, the multiplication process effectively repeats each symbol sample 400 times. In the time domain this

operation can be described as

$$y\left(m=\frac{n}{L}\right)=x\left(\frac{n}{L}\right)\otimes h(m)$$

where $h(m)$ is a length $L = 400$ Boxcar filter. In the frequency domain, the spectrum of the 50 ksps signal sampled at 20 Msps is the spectrum of the 50 ksps signal with 399 images multiplied by

$$H\left(e^{j\omega}\right)=\frac{\sin(\pi f)}{L\sin\left(\frac{\pi f}{L}\right)}\approx\frac{\sin(\pi f)}{\pi f}\qquad -\frac{f_s}{2}\left(\frac{L}{f_s}\right)<f\leq\frac{f_s}{2}\left(\frac{L}{f_s}\right)$$

This 400-tap Boxcar frequency response is shown in Figure 45.

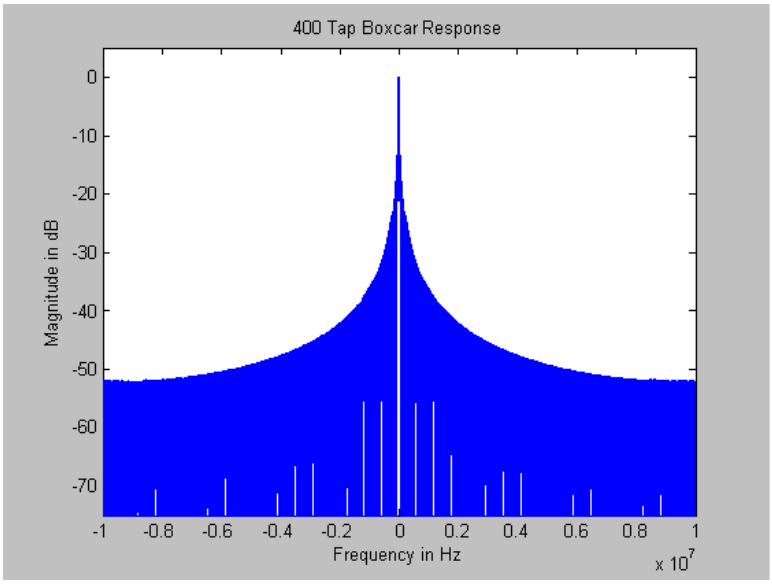

FIGURE 45. Frequency response of 400-tap Boxcar filter.

One of the side effects of this filter response is the inducing of cyclo-stationary features that reduce the entropy of the chaotic sequence. Therefore, a 79-tap halfband lowpass filter was designed with a passband edge at 4.4 MHz and stopband at 5.6 MHz. Since the design is a halfband, it is guaranteed that all even-order coefficients except for the center tap will be 0.

```
bh=firpm(78,[0, .44, .56, 1],[1,1,0,0]);
bh(2:2:78) = 0;
bh(40) = .5;
```

The filter response for the implemented 20-bit resolution is shown in Figure 46.

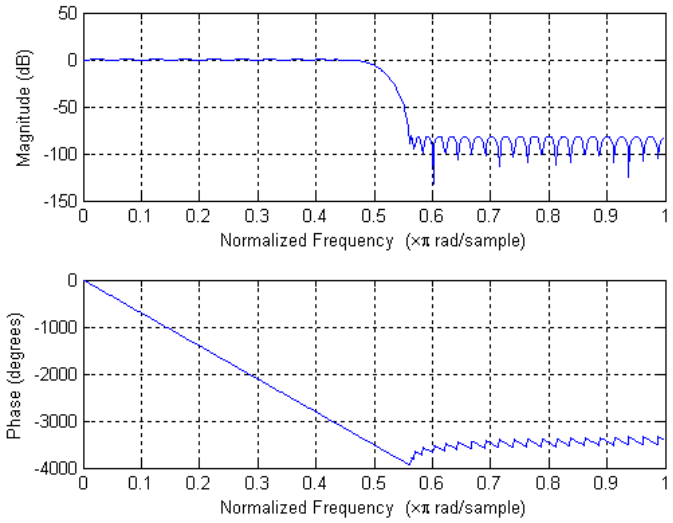

FIGURE 46. Normalized frequency response of spectral limiting lowpass filter.

The implementation of this lowpass filter follows a similar CSD replacement of hardware multipliers by static shift-additions, leading to the filter topology shown in Figure 47.

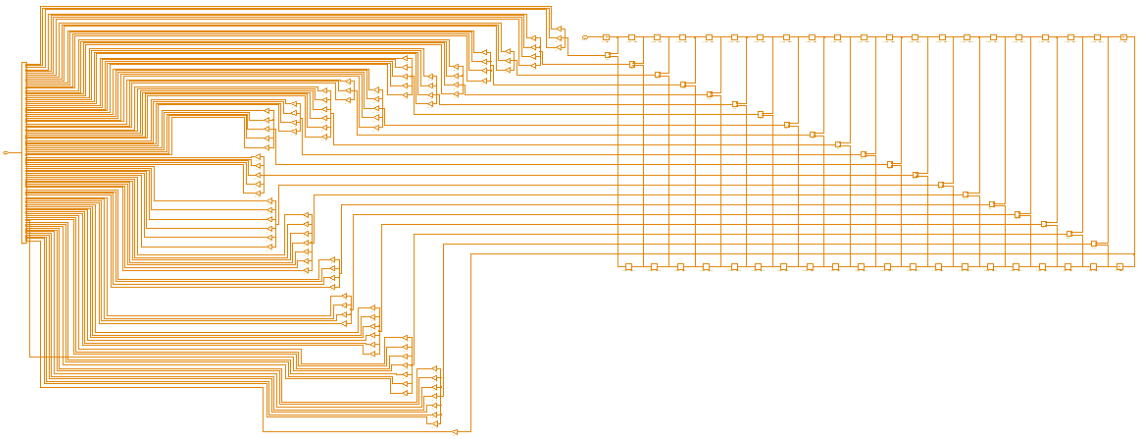

FIGURE 47. Filter topology for folded lowpass CSD implementation with pre-adds.

Using a CSD implementation for the lowpass filter replaces 21 hardware multipliers with 97 shift additions for a 56% hardware savings.

3.2.4 CPSK Data Modulation

As described in section 3.2.1, the user data is formatted into QPSK symbols that modulate the chaotic spreading sequence during the entirety of the symbol period. This modulation takes place by a complex multiplication of the complex-valued data symbol with the stream of complex-valued spreading sequence values.

$$I_{out}(nT) = I_{sym}(m\tau)I_{chaos}(nT) - Q_{sym}(m\tau)Q_{chaos}(nT)$$
$$Q_{out}(nT) = I_{sym}(m\tau)Q_{chaos}(nT) + Q_{sym}(m\tau)I_{chaos}(nT)$$

where T is the chaotic spreading sequence period and τ is the duration of a symbol ($\tau \gg T$). This multiplication may be trivially implemented using a switch-based topology, where the complex-valued data symbols are mapped to $\{(-1,-1),(-1,1),(1,-1),(1,1)\}$ as shown in Figure 48; note that this mapping results in a static gain of $\sqrt{2}$ that must be considered in subsequent stages.

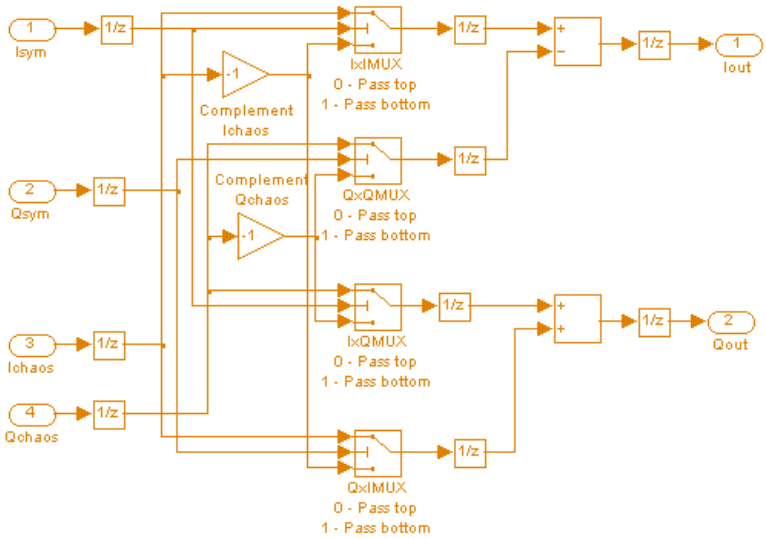

FIGURE 48. Switch-based complex multiplier.

A small amount of intersymbol interference (ISI) occurs since the data symbols are treated as binary values that switch discretely at symbol boundaries. A pulse-shaping filter was designed initially to reduce ISI, yet the induced correlation destroys the autocorrelation properties of the chaotic sequence. An exemplary histogram of a pulse-shaped data sequence modulated with the chaotic sequence (ideally Gaussian distributed) is shown in Figure 49.

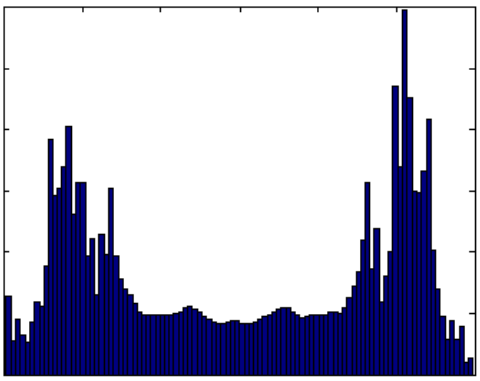

FIGURE 49. Chaotic waveform histogram with data symbol pulse-shaping.

The relatively minor effects of ISI are much preferred to the impact on the statistical characteristics of the chaotic waveform.

3.2.5 Transmit Frequency Plan Implementation

At the output of the spectrum limiting lowpass filter, the intermediate waveform has a bandwidth of 10 MHz and sample rate of 20 MHz. To complete implementation of the transmit frequency plan, a series of transformations are performed to increase the sample rate and translate to a 70 MHz IF; a series of polyphase multirate filters[136, 137] prior to handoff to an interpolating DAC[138] operating at 240 MHz that is impedance matched to the RF upconversion stage. That 70 MHz IF signal is block upconverted to a programmable 2.4 GHz ISM-band center frequency using an off-the-shelf Cross Technologies 2015-25 upconverter[139, 140]. One consideration for the RF output stage of the chaotic waveform transmitter is the effects of HPA non-linearity, or equivalently, the peak-to-average power ratio (PAPR) of the signal and resulting HPA backoff. The digitally generated chaotic sequence, prior to any filtering or other operations, has a PAPR of $20 \log_{10} \sigma_{trunc}$, or approximately 12.73 dB. The incorporation of interpolation filtering is expected to lower the maximum peak value slightly, validated by the simulated PAPR of 12.48 dB. Both of these PAPR values are significantly larger than that of most communications waveforms; two ways of rationalizing the use of this high PAPR signal are the proven maximal entropy characteristics and the fact that nonlinearities caused by insufficient HPA backoff occur in direct proportion to the decaying tails of a Gaussian distribution. Further, comparing the chaotic waveform with multi-carrier (and therefore higher PAPR) communication systems like OFDM shows better potential for multiple access communications. Later chapters will demonstrate that the PAPR of chaotic signals is an upper bound for all practical communication systems, since the separation performance[72] of overlapping chaotic signals, due to maximal entropy of the coherent despreading sequence, is optimal.

3.2.6 Chaotic Communications Transmitter Hardware Implementation

The transmitter architecture discussed in this section was implemented on the proprietary software defined radio (SDR) multi-chip module (MCM) shown in Figure 50.[27]

[27]The System-in-a-Package (SiP) SDR module is a 2" by 2" reprogrammable radio module containing FPGAs, DSP and ARM9 processors, A/D and D/A converters, power management, and other support functions. Significant assistance was provided by the Harris team (Dan Boritzki, Dave Browning, Nick Miller, Joe Petrone, and Ravi Varanasi) in converting the prototype chaotic communication system from an analytical/simulation construct to measurable hardware.

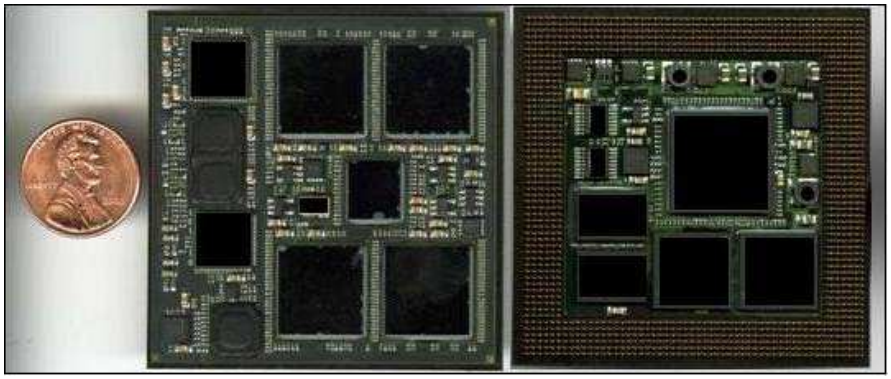

FIGURE 50. Proprietary SiP-100 destination hardware.

The primary FPGA used for the transmitter was a Xilinx Virtex-4 XC4VLX60[120], which has 64 "XtremeDSP" slices,[28] 160 block RAMs (non-parseable 1024x18 LUTs), and 53,248 standard FPGA slices, each containing a register and a four-input LUT. The transmitter was intentionally implemented to use a minimal number of hardware multipliers (more efficient in eventual ASIC implementation) and to limit the LUT usage to that required for the chaotic sequence generation; hardware usage optimization is a greater concern in the chaotic receiver discussed in Section 3.3. A summary of the hardware usage and maximum clock rates for various blocks is provided in Table 5.

TABLE 5. Digital chaotic transmitter hardware utilization.

Transmitter Component	**XtremeDSP**	**Registers**	**LUTs**	**BRAMs**
Chaotic Sequence Generator	6	6123	7502	38
Chaos Halfband Filter	0	703	8599	0
Data Source and Modulator	0	239	197	0
Digital Upconversion	8	969	7010	0
GPS Timing/Residue Calc	0	1626	762	19
TX Limiting Lowpass Filter	50	969	2049	0
Miscellaneous Logic	0	11685	5789	0
Total	64	22314	31908	57
Total Available	64	53248	53248	160

3.2.7 Chaotic Communications Transmitter Performance

The best way to discriminate between simulation models with their inherent assumptions and physical reality is to construct a measurable hardware prototype. The transmitter design and implementation described throughout this chapter was converted to a hardware prototype and evaluated for comparison to the analytical/simulation results. The Visual Elite VHDL simulations "bit-match" the SynplifyDSP models up to the DAC, indicating that the digital

[28]Each XtremeDSP slice contains one 18x18 multiplier, an adder, and a 48-bit accumulator.

signal processing chain matches the analytical predictions, leaving only the need to validate the assumptions on the DAC and RF upconversion. This section summarizes the comparison between analytical, simulation, and measured hardware performance for the chaotic communications transmitter; attention is directed more to quantify detectable features and statistical characteristics, than completing a communications link with the receiver; details describing the chaotic signal synchronization and communications link performance are discussed in Section 3.3.

3.2.7.1 Time-Domain Evaluation of Chaotic Waveform

To perform a first-order time-domain evaluation of the (non-periodic) chaotic signal, a Tektronix TLA5204B logic analyzer[141] was used to sample the signal after RF block upconversion, transmission through the communications channel, and finally downconversion to baseband at the receiver; a representative capture of the time-domain waveform, digitized at 40 Msps is shown in Figure 51. The left side of the figure contains the received chaotic signal with symbols overlayed, while the right side shows an arbitrary selection of measured samples.

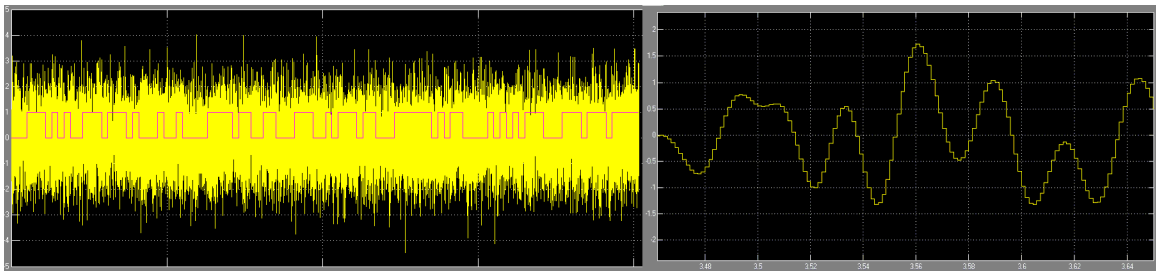

FIGURE 51. Transmitter time-domain characteristic as measured over a long (left) and short (right) time span.

These time-domain characteristics visually appear similar to the desired random signal, although the frequency domain and statistical analysis of the measured samples gives significantly better assurances that there are no significant entropy degrading waveform features.

3.2.7.2 Frequency Domain Evaluation of Chaotic Waveform

The spectral content of the transmitted signal is best characterized through the use of large Fourier transforms. The collection of responses shown in Figure 52 represent 32 consecutive 2^{16}-point DTFTs of the simulated transmitter IF output (left), compared to a 1 Mpt DTFT of the simulated RF output (center) and a 1 Mpt DTFT of the measured signal at the input and output of a noise test set (right).

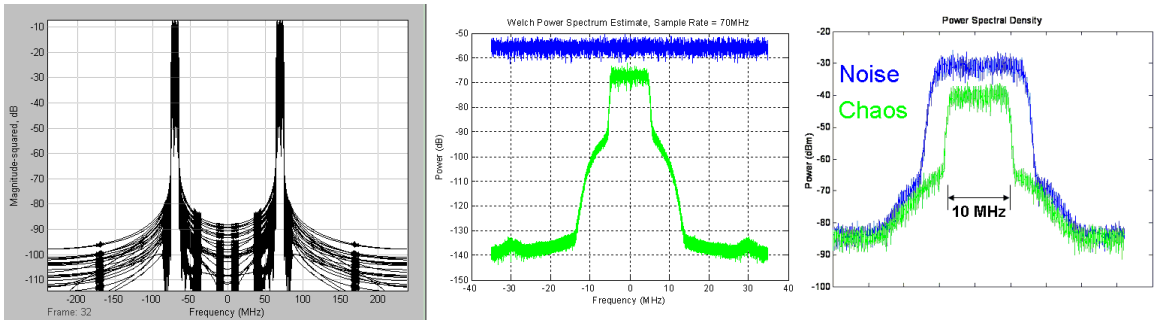

FIGURE 52. Transmitter IF spectral response, as simulated at IF (left), simulated at RF with noise additions (center), and measured at IF using a noise test set (right).

The signal content matches the analytical expectations of a flat spectral content without any spurs or other identifiable features. To validate that the RF upconversion does not induce any additional features that reduce signal entropy, the signal was upconverted and sampled by an HP model 8566B spectrum analyzer[142], combined with RF loopback testing to the prototype chaotic receiver. The baseband spectrum of the measured hardware transmitter waveform is shown in Figure 53.

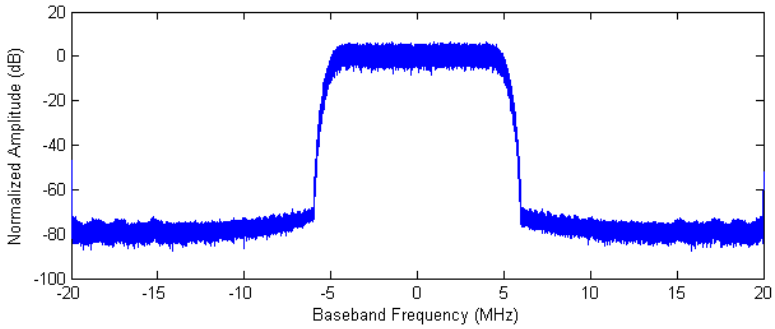

FIGURE 53. Transmitter spectral response using measured hardware RF loopback receiver samples.

3.2.7.3 Stochastic Evaluation of Chaotic Waveform

One of the most inherently beneficial characteristics of waveforms modulated with the chaotic sequence is their impulsive autocorrelation. An arbitrary collection of 1 million measured samples (40 MHz baseband sample rate) was collected for the post transmission/reception baseband signals in the RF loopback test setup. This signal was then correlated in Matlab, resulting in the emperical autocorrelation shown in Figure 54.

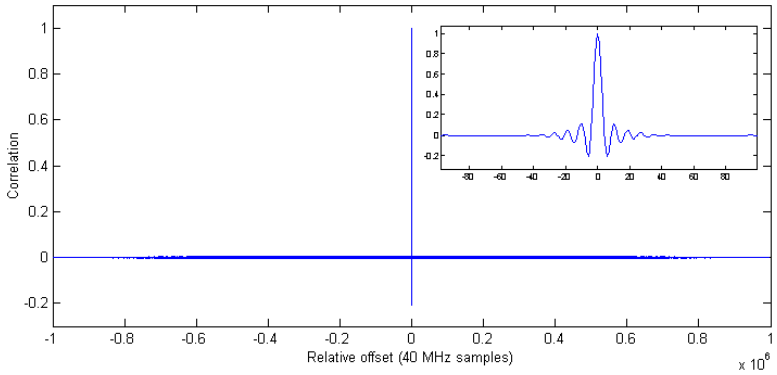

FIGURE 54. Autocorrelation of measured transmit signal via RF loopback testing.

The autocorrelations do satisfactorily show the impulsive autocorrelation characteristic; note that the correlation peak is expected to have a finite width of at least ± 1.75 chaotic spreading sequence chip durations (six 40 MHz baseband samples at 40 MHz sample rate) due to the measurement of a 10 MHz bandlimited signal at rate 40 MHz. Interpolation filtering in the transmitter is expected to increase the correlation width slightly since output values are deterministic functions of previous values (i.e. the natural impulse response of the filter). The measured RF loopback received chaotic waveform shows a correlation peak width of approximately ± 2.25 chaotic sequence chip durations (≈ 225 ns), bounded conservatively as a 10 dB reduction in the autocorrelation. The autocorrelation shows a minimum 20 dB reduction outside of ± 8 chaotic sequence chip durations.

Arbitrary collections of 1 million measured samples (40 Msps sample rate) were collected at the chaotic receiver after transmission and reception through a live communications channel for statistical analysis. The statistical characteristics quantified through the cumulant measures for 100 random trials of the pre-transmission simulated IF signal, pre-transmission measured hardware IF signal, and post transmission/reception signal are summarized in Table 6. Considering confidence intervals about the preceding moment generating function analysis for ideal AWGN samples, the chaotic waveform is virtually indistinguishable from background AWGN.

TABLE 6. Cumulant comparison between simulated, RF loopback hardware testing, and live OTA measured chaotic waveforms.

Cumulant	$N(\mu = 0, \sigma^2 = 1)$	**IF Simulated**	**RF Loopback**	**OTA Measured**
1^{st} :	0	$-7.83 \cdot 10^{-8}$	0.0003	$1.02 \cdot 10^{-6}$
2^{nd} :	1	1.0000	0.9993	0.9989
3^{rd} :	0	$-5.83 \cdot 10^{-7}$	0.0025	$-1.50 \cdot 10^{-4}$
4^{th} :	3	2.9960	2.9912	2.9892
5^{th} :	0	$-9.14 \cdot 10^{-7}$	0.0128	$-2.71 \cdot 10^{-3}$
6^{th} :	15	14.9359	14.8844	14.8853
7^{th} :	0	$5.97 \cdot 10^{-3}$	0.0467	-0.0521
8^{th} :	105	104.0925	103.3945	103.7562

The convergence of the simulated and measured statistical characteristics validates the analytical predictions of the chaotic PSK waveform as a near-ideal maximal entropy waveform for use in a coherent communication system. Moreover, the analysis, simulation, and hardware measurements of the prototype chaotic communications transmitter under time domain, frequency domain, and stochastic analysis all support the fully digital generation and transmission of chaotic signals that are indistinguishable from those created by analog chaotic communication systems mapped to quadrature Gaussian spread spectrum chipping sequences. The next step is to implement a coherent chaotic receiver constructed around a synchronized digital chaotic sequence generator.

3.3 Prototype Chaotic Communications Receiver

The traditionally difficult task in implementing a coherent chaotic communication system has been satisfactorily synchronizing the chaotic circuits at the transmitter and receiver[42, 41, 69] so that user data may be modulated, transmitted, and decoded efficiently. To date, there are believed to be no published chaotic circuit synchronization methods that are robust enough to provide the basis for a practical chaotic communications system[40, 41, 42, 44]. In fact, some[143] have questioned the suitability of traditional control mechanisms like early-late tracking loops for chaotic waveforms, while others[37] have proposed iterative channel equalization methods as a solution to maintaining a robust chaotic circuit synchronization in varying channel conditions. This section outlines a prototype coherent chaotic communications receiver, including system-level architecture overview and comparison of predicted and measured performance. Detailed analysis leading to the core chaotic signal acquisition, chaotic circuit synchronization, and generalizations of direct sequence spread spectrum receiver processing is included in Chapter 4.

A three-part paper by Kolumban, Kennedy, and Chua[5, 43, 44] was published from 1997 to 2000, exploring the role, techniques, and performance bounds of synchronization in coherent chaotic communication systems. In general, coherent chaotic receivers can recreate exact duplicates of the chaotic sample functions used at the transmitter to modulate data; noncoherent receivers lack the ability to recreate or maintain a lock on all possible chaotic state evolutions experienced at the transmitter. Kolumban's work builds on the 1990 observation by Pecora and Carroll[32] that chaotic systems can be synchronized, focusing on the need and limits that synchronization plays. One limit of this paper is the reliance on analog chaotic circuits, derived from variations of Chua's original chaotic circuits[2, 26]. The derived results significantly match those obtained in the simulation and hardware measurements for the chaotic communication system described in this dissertation, constructed using digitally generated discrete-time discrete-amplitude chaotic circuits.

3.3.1 Chaotic Communications Receiver Architecture

The general structure of an adaptive correlator-based coherent chaotic receiver requires a time and carrier phase synchronized chaotic signal to accurately despread the received chaotic signal. An exemplary architecture for the coherent chaotic receiver is shown in Figure 55.

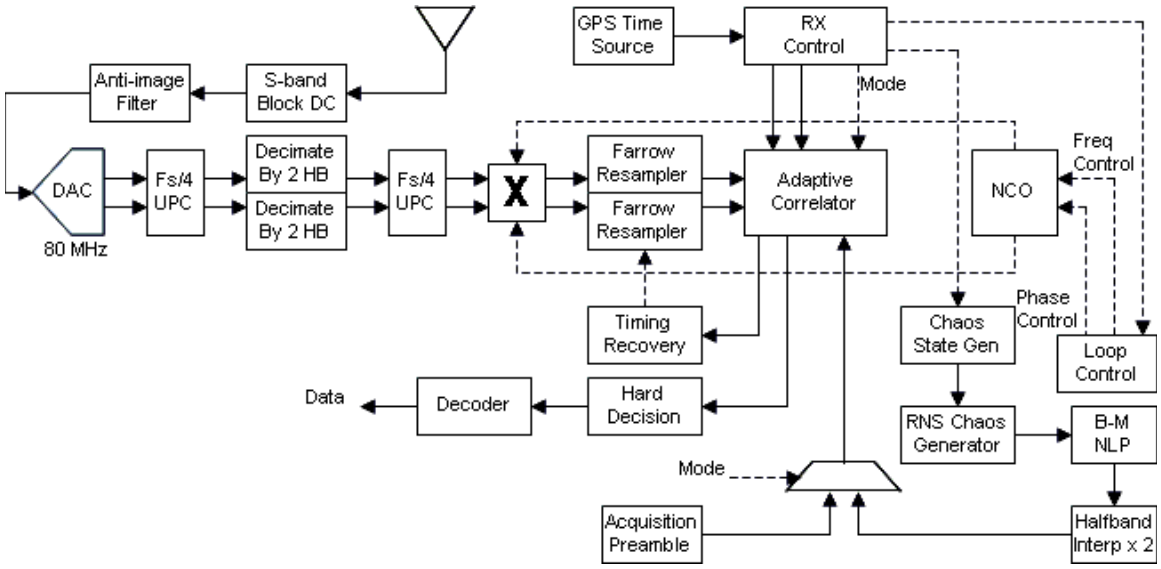

FIGURE 55. Coherent chaotic PSK receiver architecture.

Walking through the coherent chaotic receiver architecture, the received signal is block downconverted[144, 145] to a 70 MHz IF, subsampled at 80 MHz, and converted to a ± 5 MHz baseband signal at a sample rate of 40 MHz. Phase and frequency adjustment are introduced by an NCO coupled to phase and frequency error detectors. Timing adjustments are applied by a combination of timing sources including adjustable 4-tap delay lines (not shown) and

a Farrow architecture continuous phase resampling filter. This phase and time synchronized signal is then despread using the internal replica of the digital chaotic sequence generator, which is interpolated to a sample rate of 40 MHz. Error calculators, similar in nature to DS spread spectrum early/late detectors, are applied at a sample rate of 100 ksps, or two soft decisions per data symbol. These soft decisions are combined and decoded to provide an output data stream, ending the demodulation process.

Of particular interest in this section are the implementation trades used for chaotic signal reception and processing. Detailed evaluation of novel signal processing structures, including chaotic signal acquisition via an adaptive correlator, chaotic circuit synchronization, and generalizations of direct sequence spread spectrum receiver processing is included in Chapter 4.

3.3.2 Chaotic Receiver Timing Control

The impulsive autocorrelation of the chaotic waveform necessitates a highly robust timing control methodology that ensures the relative delay between the received chaotic waveform and the internally generated chaotic sequence is less than one spreading chip duration (100 ns). Moreover, the ability to time synchronize the received and internally generated chaotic signals within approximately 0.1 spreading chip durations (10 ns) is prefered to reduce receiver implementation loss and susceptibility to time tracking loop errors. Timing control was broken into a timing error detector, three discrete timing quantizations as well as a synchronized symbol timing clock. Coarse timing synchronization is implemented using three chaotic sequence generator controls, permitting a static jump to any chaotic sequence state within one clock cycle or a brief pause/acceleration to enable arbitrary time tracking; these coarse timing steps can synchronize the internal and received chaos signals to within $\frac{1}{10MHz} = 100$ ns, which is insufficient for robust demodulation. Medium timing synchronization continues with a set of four adjustable delays at the baseband sample rate of 40 MHz, permitting synchronization to within 25 ns, or one-fourth of a chaotic sequence chip period. Refining this timing synchronization even further, a Farrow architecture continuous phase resampling filter is placed in the receive signal path to permit sub-sample resampling, with a precision that is well beyond the accuracy of the shared GPS reference synchronization.

3.3.2.1 Timing Error Detection

The fundamental design for timing error detection is that of a direct sequence spread spectrum early-late detector. Along with the primary 'prompt' despreader, a pair of additional despreaders with relative delays of $\pm\frac{1}{2}$ chaotic sequence chips ($\pm$50 ns) was constructed and continually evaluated for timing drifts in the correlation peak. The early-late detection

mechanism produces an error signal corresponding approximately linearly to the correlation peak time drift[29]; a notional depiction of the first-order early-late error detection is shown in Figure 56.

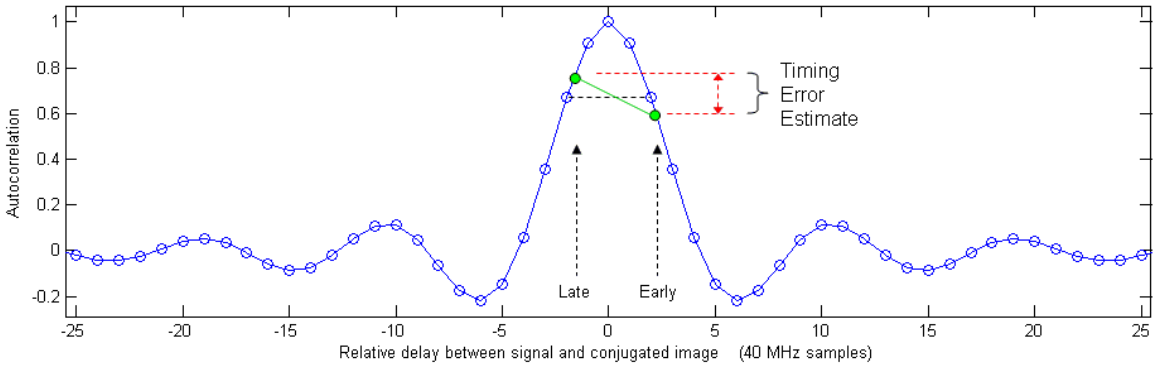

FIGURE 56. Notional depiction of static first-order early-late time error detection.

The loop bandwidth of the first-order time tracking loop must be chosen sufficiently large to mitigate realistic timing drifts of physical motion between transmitter and receiver. Since the despreader soft decisions are produced at a 100 kHz rate, which is equivalent to the loop update rate (LUR), and realistic ground platform motions are constrained to approximately 100 $\frac{m}{s}$, then the loop must be able to maintain track with 333 ns of signal drift per second, or 3.33 chaotic sequence chip durations per 100,000 timing loop updates. Additional to this ground platform motion estimate for time tracking is the timing drift of internal oscillators, which is on the order of 10 ppm, or 1 chaotic sequence chip duration per 100,000 loop updates. Note that a much wider loop bandwidth (approximately 8 times nominal) is used during time tracking of the preamble and that error saturation blocks are included to prevent timing error outliers from breaking time synchronization. Mechanisms were also implemented to clear the current error accumulation whenever an integer sample boundary is crossed as well as provide signal lock indication.

3.3.2.2 Timing Control

The output of the timing error calculator is provided to a second timing block that accumulates and parses the time error to various correction blocks throughout the receiver. A notional view of the timing control loop is shown in Figure 57.

[29]In severe multipath environments, the effective correlation peak becomes distorted and can give faulty error estimates. Various post-correction techniques exist for implementing more robust early-late detection, including extension to additional early-early and late-late despreaders as well as variable delays in the early-late detection. Many of these techniques have been published with respect to precision GPS reception[146, 147, 148, 149].

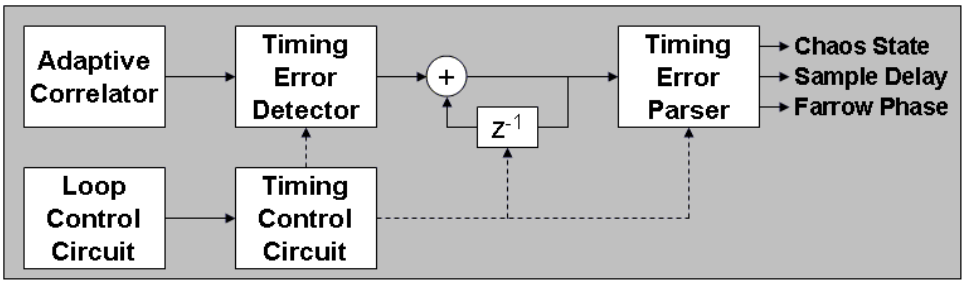

FIGURE 57. Top-level timing control loop implementation.

This block combines the real-time tracking time error estimate with the output of the timing loop initialization estimates, which are activated at the conclusion of acquisition processing. Error accumulation inside the Farrow resampler ranges from $[-0.1, 1.1]$ in order to induce hysteresis at integer sample delays and generate the Farrow resampler control;[30] even though the "fractional" accumulation ranges from $[-0.1, 1.1]$ and the Farrow resampler is designed for fractional delays in the $[0, 1]$ range, less than 0.05 dB of additional receiver implementation loss is incurred to eliminate integer sample boundary ambiguities. Whenever the "fractional" accumulation exceeds the range $[-0.1, 1.1]$, an integer value is passed to the second accumulation stage for integer numbers of 40 MHz baseband samples; this second accumulation is broken into a count that corresponds to integer delays in the received chaotic signal, a delay that corresponds to integer delays in the internally generated chaotic signal, and a symmetric delay control that time aligns the symbol clock with the synchronized pair of chaotic signals. Whenever this "fractional" accumulator rolls over, a pulse is generated to clear the first-order time error estimator accumulation. Note that the effective count of the integer accumulation is $[-4, 4]$ relative delays. All integer delays are implemented with dynamic shift registers at the 40 MHz sample rate. Whenever this second accumulation reaches the boundary of the [-4,4] accumulation range, another pulse is generated and distributed to the internal chaotic sequence generator to accelerate/decelerate the sequence by one 10 MHz sample (equivalent to four 40 MHz samples); the latency between kicking the chaotic sequence generator and fully implementing the time alignment produces 1-2 non-optimal soft symbol estimates that may be easily eliminated with forward error correction.

3.3.2.3 Farrow Resampler

A Farrow resampling filter[150, 151, 152] was implemented to provide a continuously variable fractional delay in the received chaotic signal; the general concept behind the Farrow

[30]This hysteresis is intentionally created to mitigate the effects of repeated hopping across integer sample time boundaries, which can require up to fifty 40 MHz sample periods in processing latency.

resampler is a multiphase interpolation filter that accepts an input control on $[0, 1]$ and implements the fractional delay associated with the control; filter coefficients were chosen via polynomial curve fits and then conversion to canonic signed digit shift-add implementation. The implemented Farrow filter used six polyphase segments and fourth-order polynomial fits to coefficients for a worst-case SFDR of greater than 80 dB.[31] The resulting topology and response for the Farrow resampling filter, implementing precise delays to within 0.01 samples (250 ps), is shown in Figure 58.

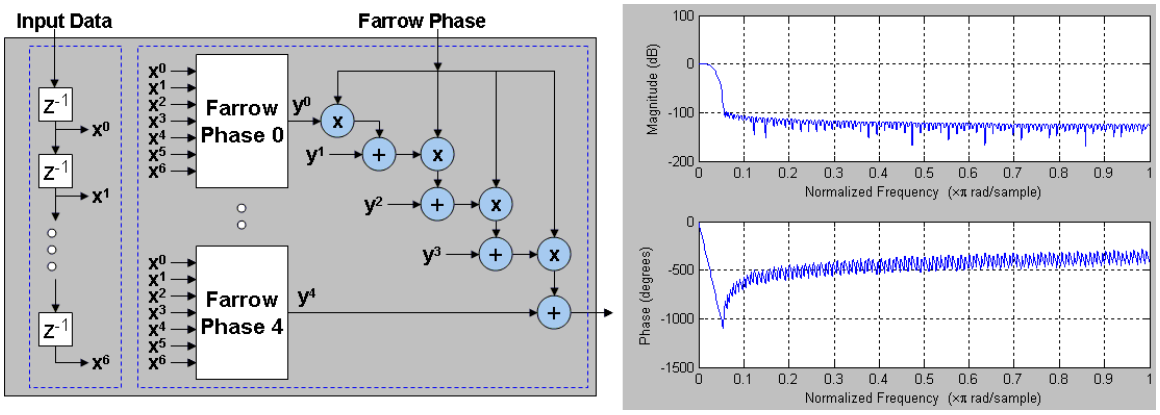

FIGURE 58. Block diagram of Farrow resampling filter.

3.3.2.4 Symbol Timing

Symbol timing synchronization falls directly out of the chaotic sequence controls, with the symbol timing clock derived from the enable structure of the chaotic sequence generator. As implemented on the transmitter, a ring generator that counts two hundred 20 MHz enables is embedded in the chaotic sequence generator and level detected to produce a one bit symbol clock. This 50% duty cycle symbol clock is positive edge detected and distributed to the remainder of the receiver for coordinating the integrate-and-dump cycles of the despreader, provide enables to the tracking loops, and output data. The symbol clock was calibrated during simulation to account for the processing latency of the halfband filters and despreader; slight variations of the symbol clock occur in the timing control loop, where relative drifts between the internal and received chaotic signals within a spreading chip duration are translated to integer numbers of 40 MHz delays and compensated.

3.3.3 Phase and Frequency Control

Equally important to accurate demodulation of the chaotic waveform is a robust phase and frequency tracking methodology that can detect and compensate for phase or frequency

[31] Slight modifications to the Matlab code implied by Fred Harris' paper[150] were implemented as a result of personal correspondance with Harris and David Chester of Harris Corporation.

drifts during transmission. A similar framework to early-late detection was used for estimating phase errors relative to the ideal QPSK constellation points and correcting via a second-order tracking loop. These tracking loops then feed a numerically controlled oscillator (NCO) that modifies the phase of the received chaotic signal via a complex multiplier. A top-level diagram of the implemented phase and frequency tracking loops are shown in Figure 59.

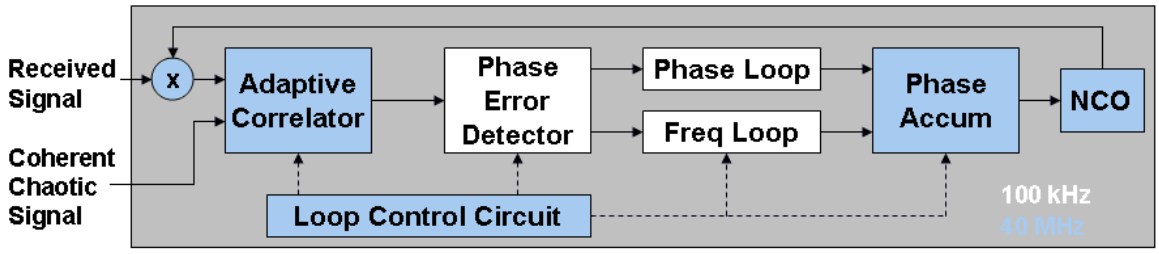

FIGURE 59. Top-level diagram of phase and frequency loops.

3.3.3.1 Phase Error Detection

A pair of phase error detectors were used to evaluate despread soft symbols for drifts and provide an error signal to the tracking loops. The first phase detector is a four-quadrant coarse angle ($\frac{\pi}{32}$) estimator that is used during the initial acquisition preamble to quickly lock onto the phase and frequency of the received signal (wider loop bandwidth). The second phase detector is a one-quadrant fine-angle ($\frac{\pi}{256}$) estimator that drives the tracking performance of the phase and frequency loops. Each error detector is implemented as nonlinear processor (NLP) architectures, similar to the Box-Muller transformation, mixing low-order Taylor series approximated arithmetic with binary addressed LUTs; most significant in the designs is a multiplicative inverse block that estimates the inverse tangent operation as

$$\Phi(x) = \tan^{-1}\left[\frac{Im(x)}{Re(x)}\right] \approx \frac{Im(x)}{Re(x)} = Im(x) \cdot Re(x)^{-1}$$

where x is a soft symbol estimate. The phase error for these error detectors was verified relative to sinusoidal inputs and is sufficient accurate over 10 binary orders of magnitude.

3.3.3.2 Phase and Frequency Loop Implementation

The phase and frequency loop was initially chosen to have a loop bandwidth of $B_L = 1024$ Hz, implemented as a standard second-order digital Costas[153] loop with lead-lag topology[154]. For a spread-spectrum signal, the optimized input is at a notional quarter-full-scale input level, which is then phase adjusted and input to an integrate and dump despreader. The phase error signal is then provided to the second-order loop for phase and frequency tracking, and finally to an NCO for conversion back to a phase correction. The loop update rate (LUR) was chosen

consistent with the 100 ksps soft symbol rate, along with the baseband sample rate of 40 MHz and a chosen damping factor of $\frac{\sqrt{2}}{2}$. These choices drive the lead (K_F) and lag (a_2) terms to

$$\omega_n = \frac{2B_L}{\zeta + \frac{1}{4\zeta}} = \frac{2 \cdot (1024\,\text{Hz})}{\frac{\sqrt{2}}{2} + \frac{1}{2\sqrt{2}}} \approx 1931\,\frac{\text{rad}}{\text{sec}}$$

$$K_F = \frac{4\zeta\omega_n}{(2\pi)\frac{SR}{UR}} = 2.173 \qquad a_2 = \frac{\omega_n}{2\zeta} = 1365$$

Experimental optimization and hardware efficiency implementation adjustments to the standard second-order loop parameters resulted in a loop bandwidth of approximatel 1200 Hz; this slightly wider bandwidth is believed to partially result from the nonstationary energy content of the soft symbol decisions at the output of the despreader. An additional pair of proprietary tracking loops leveraging known nonlinearities in the chaotic waveform were also constructed and validated in simulation; these loops performed better in simulation, both to CW interference and at varying noise levels, but were discarded in the final implementation due to failing unconditional loop stability tests.

3.3.4 Data Symbol Operations

Similar to any spread spectrum signal, the incoming baseband samples are despread using a conjugated internal replica of the original spreading sequence, resulting in a phase-coherent accumulation of energy (collapsed spectral bandwidth) that represents a data symbol. In the case of a chaotic waveform, the despread signals have non-stationary amplitude, defined by the variance of a short-run integration of a Gaussian sequence. After converting the accumulated value to a hard symbol decision, symbols are decoded into data bits; adjustments may be made based on known preambles and frequency inversion disambiguity bits. Additional data integrity may be ensured by the use of forward error correction (FEC), interleaving/deinterleaving, and periodic ambles like baseband injected pilot carriers[155].

3.3.4.1 Symbol Decisions

The basic concept for soft symbol decisions for a chaotic waveform is identical to that of a direct sequence spread spectrum communication system: a short-term energy integration of the received signal multiplied with a time-synchronized conjugated internal replica of the original spreading sequence. The practical goal is to increase the data throughput by lowering the spreading ratio, which in turn increases the ratio of symbol variance to symbol mean. The chosen spread ratio of 200 and oversampling to 40 MHz results in 800 baseband samples per 50 kHz soft symbol decision. The oversampling reduces the number of distinct samples by

approximately 4x, making the effective symbol decision, assuming perfect time alignment, a function of 200 draws on a zero-mean Gaussian PRNG.

Let X and Y represent a sequence of draws from the internally generated and externally received chaotic signals, and N be a bandlimited white noise sample sequence representing ambient AWGN. If X and Y have variances (signal power) σ_x^2 and σ_y^2, respectively, than a parameter α may be defined as the relative scaling factor between the received chaotic signal and a normalized stream of internally generated chaotic sequence values (standard Normal distribution). For any time-synchronized sample of X and Y,

$$y_i = \alpha x_i = |\alpha| e^{j\phi(\alpha)} x_i \qquad \text{and} \qquad \sigma_y = |\alpha| \sigma_x$$

The stationary expectation of the despread signal energy over a symbol duration T, adjusted for the oversampling rate, is

$$E[E_{sym}] = |\sum_{k=0}^{T} (x_k - \mu_x)^* \cdot (y_k - \mu_y)| = |\sum_{k=0,\, k \equiv 0 \, mod 4}^{799} x_k^* \cdot (\alpha x_k)| = 200 |\alpha| \sigma_x^2$$

This stationary symbol energy estimate is an approximation resulting from the Central Limit Theorem (CLT), but it does not directly account for the statistical distribution of the signal. Treating the symbol estimation process as an accumulation of $\frac{800}{4} = 200$ squared Gaussian samples (α is a complex constant), we obtain a Chi-squared distribution with 200 degrees of freedom for the symbol estimate. This distribution has the probability density function,

$$f_{E_{sym}}(u) = \frac{u^{99} e^{-\frac{u}{2}}}{2^{100}\, 100!}$$

as depicted graphically in Figure 60.

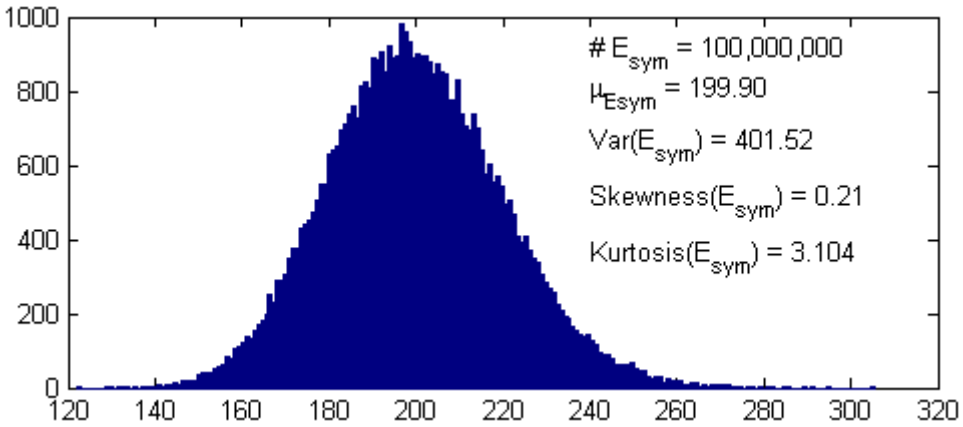

FIGURE 60. Histogram of 100M Chi-square soft symbol estimates.

Note that the right tail of the distribution carries more outliers, indicating that short term averages of the signal are more likely to show larger deviations to the right than the left. The distribution is sufficiently close to that of a true Gaussian distribution as estimated with the CLT and verified by rough agreement of the first four cumulants to generate symbol error rate estimates with standard AWGN estimators. Returning to the content of the received chaotic signal, the signal sample sequence may be written as[32]

$$s_i = y_i + n_i = \alpha x_i + n_i$$

The noise contribution to the soft symbol estimate is strictly additive to the received signal, resulting in an accumulation of despread samples with statistical distributions related to a modified Bessel function. More precisely, the probability density function for the multiplication of two independent zero-mean Gaussian random variables is

$$p(z) = \frac{1}{\pi\sigma_x\sigma_n}\frac{z}{|z|}K_0\left(\frac{|z|}{\sigma_x\sigma_n}\right)$$

where K_0 is a modified Bessel function of the second kind

$$K_0 = \int_0^\infty \frac{\cos(zt)}{\sqrt{t^2+1}}dt$$

and $z_k = x_k \cdot n_k$. The Bessel function integral is clearly symmetric about 0, making $\mu_z = 0$, and the numerically approximated variance is $\sigma_z^2 = \sigma_x^2\sigma_n^2$. A numerically generated histogram for 10M multiplied independent standard Normal samples is shown in Figure 61.

[32]Expansion of the bandlimited noise signal to account for the RF filter bandwidths in a practical receiver makes direct comparison of the received signal power and the noise power somewhat inaccurate. Adjusting the noise bandwidth in the present system to approximately 12.5 MHz as opposed to 10.0 MHz, the effective minimum receiver implementation loss is approximately $10\log_{10}\frac{12.5MHz}{10.0MHz} \approx 1.0$ dB. This receiver loss will be addressed more concretely with regard to discussion of improvements to the hardware prototype coherent chaotic communication system; for the moment, the received signal and noise bandwidths are treated as identical.

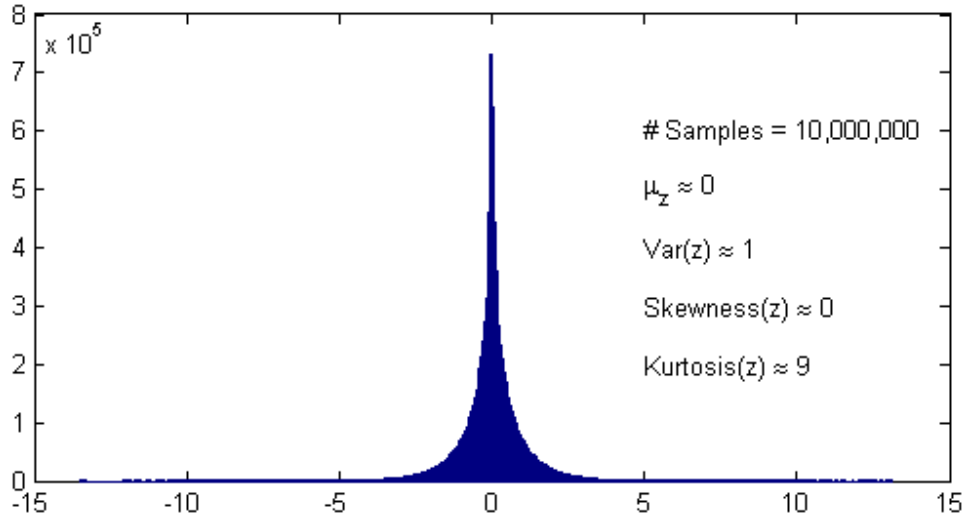

FIGURE 61. Histogram of 10M independent zero-mean Gaussian random variable products.

After accumulating 200 of these independent noise samples, the overall distribution is very close to a Gaussian distribution with zero-mean and variance $200\sigma_x\sigma_n$. A histogram accumulated over 100M independent standard Normal products (decimated to 50K independent accumulations) is shown in Figure 62.

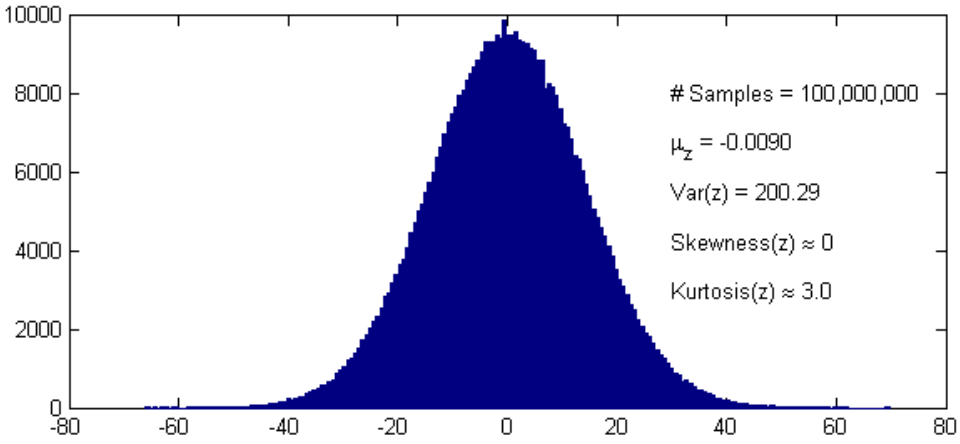

FIGURE 62. Histogram of 100M independent standard Normal products.

Both simulations and hardware measurements have been used to validate the suitability of these basic approximations for the chaotic waveform. A summary estimate of the noise energy integrated over a symbol duration is

$$E[N_{sym}] = |\sum_{k=0}^{T}(x_k - \mu_x)^* \cdot (n_k - \mu_n)| = |\sum_{k=0,\, k\equiv 0 \mod 4}^{799} x_k^* \cdot n_k| = 200\sigma_x\sigma_n$$

so that the overall effective[33] soft symbol estimate is

$$E_{Sym} = 200|\alpha|e^{j\phi(\alpha)}\sigma_x^2 + 10\sqrt{2\sigma_x\sigma_n}Z(0,1)$$

where $Z(0,1)$ is a standard Normal random variable. By noting that automatic gain control and the binary representation of the digitized samples can be chosen such that $\sigma_x = \sigma_n = 1$, the only effective free parameter remaining is α, which is equivalent to a measure of the carrier to noise ratio, $|\frac{Y}{N}| = \frac{|\alpha|\sigma_x}{\sigma_n} = \frac{|\alpha|\sigma_x}{\sigma_x} = |\alpha|$. Assuming active phase tracking, the range of possible values for $\phi(\alpha)$ are

$$\phi(\alpha) \in \{e^{j\frac{\pi}{4}}, e^{j\frac{3\pi}{4}}, e^{-j\frac{3\pi}{4}}, e^{-j\frac{\pi}{4}}\}$$

These soft symbols then create a signaling constellation for soft symbols as shown in Figure 63.

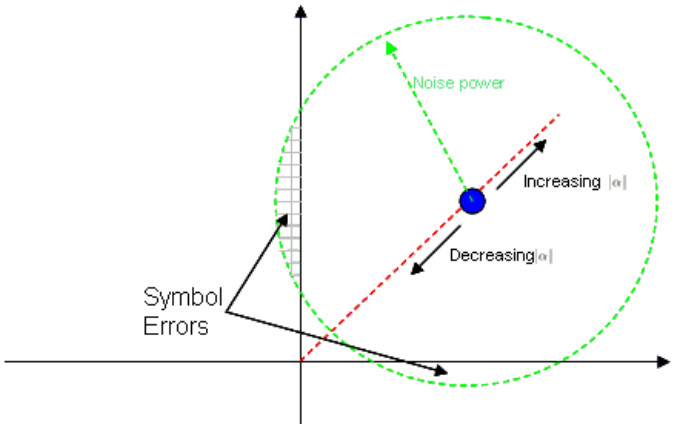

FIGURE 63. Soft symbol composition with noise and signal energy.

Soft symbol decisions are performed via quadrature sign selection on the resulting constellation,[34] effectively projecting the resulting constellation point onto the real and imaginary axes. As expected, when the ratio of signal energy and noise energy decreases, the probability of obtaining a symbol error increases. The theoretical symbol-error rate performance and comparative simulation/hardware measurements will be developed in Chapter 4. Various improvements to the basic spread spectrum symbol operations for a chaotic waveform that increase the effective signal-to-noise ratio and BER performance, are also included in Chapter 4.

[33]This estimate includes an adjustment for the oversampling rate.
[34]A relatively simple symbol synchronization loop is implemented so that two successive 100 KHz soft symbol estimates are added to create the 50 KHz data symbol estimates, prior to hard decisions.

3.3.4.2 Preamble, Disambiguity Bits, and Symbol Decoder

In addition to the soft symbol creation and decision process, three additional logical conditions are inserted into the data sequence to aid accurate reception of the chaotic sequence modulated QPSK signal. A preamble sequence of 200 data symbols provides the receiver with a sufficient duration to perform acquisition ($\approx 500\ \mu s$), loop initialization ($\approx 50\ \mu s$), and tracking lock ($\approx 1ms$), plus a wider range for harsh transmission environments. This preamble is 200 repeated "00" symbols, which is mapped to the first quadrant of the receiver constellation, and effectively creates a chaotically modulated CW tone. At the conclusion of the preamble, two disambiguity symbols are appended to detect and correct whether frequency inversion has occurred. Finally, a simple symbol decoder was constructed to invert the Gray code mapping process implemented at the transmitter. The inversion process requires a small handful of digital logic gates, followed by interface logic to send the data decisions out of the receiver.

3.3.5 Chaotic Waveform Demodulation

After performing acquisition processing, loop initialization, and initial tracking operations, the primary focus of the coherent chaotic communication system is the data demodulation performance of the receiver. As described previously, the received soft symbol estimate may be reduced to independent samples having statistical distributions $N(|\alpha|R, \sqrt{(1+2|\alpha|^2)R^2})$, where $R = \frac{800}{4} = 200$ is a constant. The probability of a QPSK symbol error is equivalent to an integration of the probability of noise pushing the symbol estimate across one of the coordinate axes. In addition to the theoretical symbol error rate, uncertainty in the phase and frequency tracking loops and non-stationary symbol energies also modulate the real symbol rate. This section begins with an analytical development of the predicted symbol error rate under ideal AWGN conditions, continues with adjustments for tracking errors, and concludes with an evaluation of non-stationary energy in the chaotically modulated data symbols.

3.3.5.1 Predicted Symbol Error Rates of Chaotic Phase Shift Keying

First considering a BPSK constellation, a symbol error rate may be estimated from the likelihood that a random draw from the soft symbol distribution will be negative. Under the assumption of well behaved tracking loops and a stationary input signal power envelope, the probability of a symbol (bit) error[35] is

[35]The reduction of the high-order Chi-square distribution to the approximated Normal distribution provides a worst case error estimate and therefore a lower bound on BER performance.

$$P_e = P(z < 0) \quad z \equiv N(200|\alpha|, \sqrt{200 \cdot (1 + 2|\alpha|^2)}^2)$$

$$P_e = Q\left(\frac{z - 200|\alpha|}{\sqrt{200 \cdot (1 + 2|\alpha|^2)}}\right) = \frac{1}{2}\, erfc\left(\frac{200|\alpha|}{\sqrt{400 \cdot (1 + 2|\alpha|^2)}}\right)$$

where $Q(u)$ is the common Q-function. For small $|\alpha|$, the P_e approaches 50% since z approaches a zero-mean Gaussian random variable; for large $|\alpha|$, the P_e approaches 0% since the non-zero mean z requires drawing an increasingly large statistical outlier to cross to the negative boundary. Both of these observations are consistent with the expected performance of any BER estimator. As an intermediate example, the expected P_e with a -20 dB spread signal power ($|\alpha| = 0.1$) is 8.1%, which is consistent with a +3 dB despread signal power.

3.3.6 Chaotic Receiver Hardware Utilization

To measure the performance of a real coherent chaotic communications link, a single channel receiver was constructed using the model based synthesis approach in Synplify DSP. This process started with a validated gate-level Synplify model used to simulate performance before converting to VHDL. A collective team effort then took this Synplify model and implemented an FPGA wrapper for interfacing to the receiver. The chaotic receiver hardware utilization shown in Table 7 represents a non-optimized implementation; in particular, minimal additional control logic may be added to reuse the despreader multipliers from the pool of adaptive correlator resources since acquisition and demodulation represent distinct receiver modes. The large size of the interface and control logic is primarily a function of the multi-level test port muxes used for extracting measured data from intermediate processing points within the receiver.

TABLE 7. Prototype digital chaotic receiver hardware utilization.

Receiver Component	XtremeDSP	Registers	LUTs	BRAMs
Adaptive Correlator	27	7020	8135	37
Chaos Halfband Filter	0	703	8599	0
Chaotic Sequence Generator	6	6123	7502	38
Despreader	13	1883	2261	0
Digital RF Front End	0	423	1143	0
Farrow Resampling Filters	7	1192	6864	7
GPS Timing/Residue Calc	0	1066	609	0
Interface and Control Logic	1	8927	1834	2
Numerically Controller Oscillator	3	70	267	0
Phase and Frequency Tracking	5	2294	2932	2
Symbol Normalization/Decoder	2	990	673	1
Time Tracking	0	1454	1200	0
Total	64	32145	42019	87
Total Available	64	53248	53248	160

3.4 Prototype Chaotic Communications Validation

The system-level design and implementation for the prototype chaotic PSK communication system described in this chapter represents what is believed to be the first practical coherent chaotic communication system. The transmitter and receiver designs implement identical digital chaotic circuits that may be synchronized as accurately as the available timing references at each end; combined they were harnessed to construct a practical coherent chaotic communications link. Each end of this simplex link was implemented in an FPGA and deployed using completely independent hardware platforms to test as an over-the-air (OTA) link in the 2.4 GHz ISM band. Specific details of the testing regime, lab setup, higher level protocol functionality, and end application(s) are not included in this document to protect proprietary interests, although a basic block diagram of the test setup is shown in Figure 64.

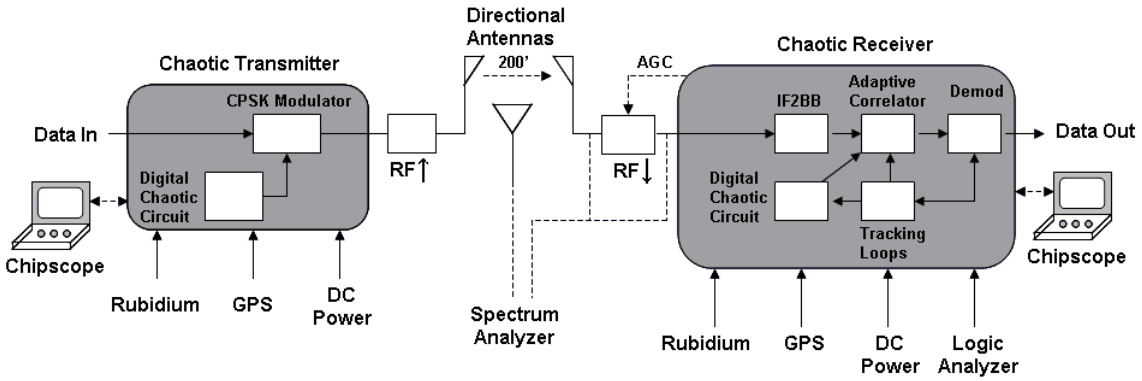

FIGURE 64. Block diagram of prototype chaotic communications test setup.

This test setup was implemented on top of Harris Corporation Building 102 as photographed in Figure 65.

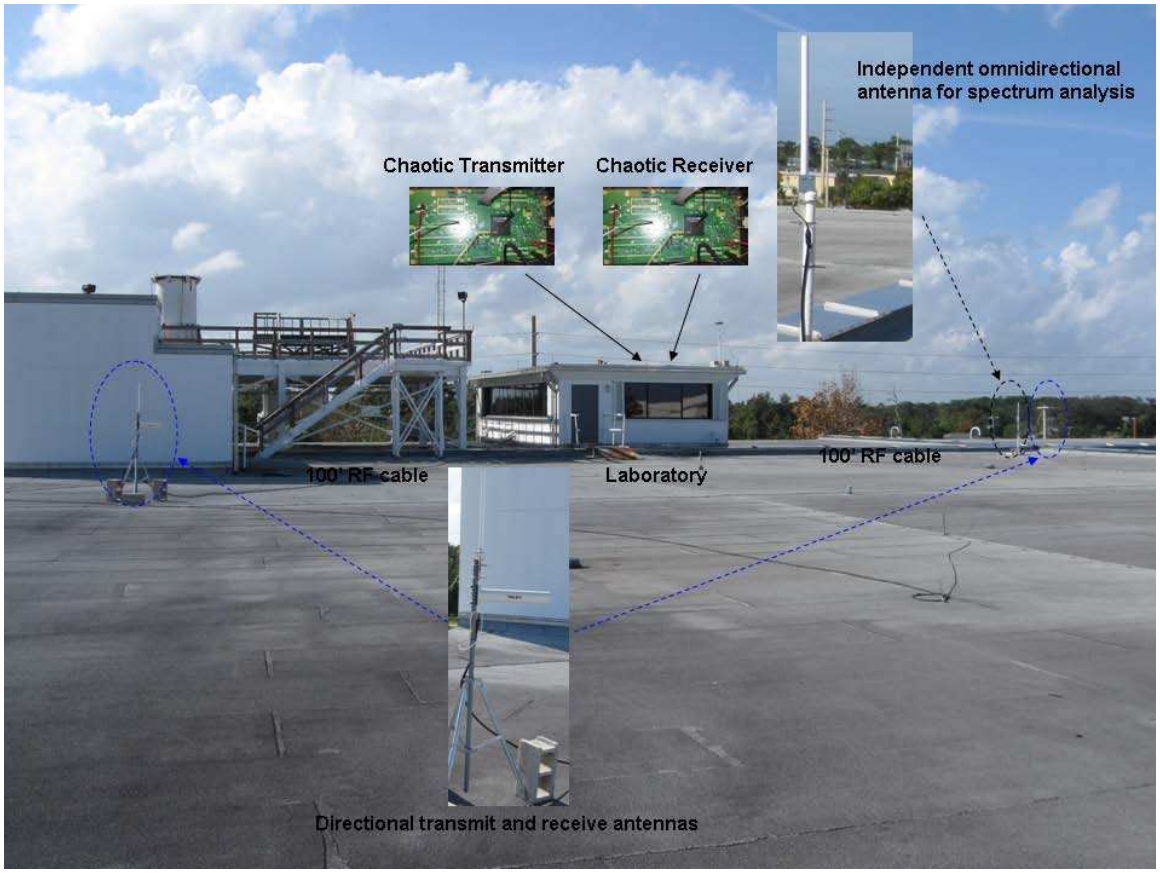

FIGURE 65. Photograph of prototype chaotic communications setup.

Detailed analysis that presents the mathematical foundation of coherent chaotic communications, along with simulated results validated by hardware measurements for the prototype chaotic communication system presented in this chapter are presented in chapter 4.

Chapter 4: Coherent Chaotic Communications Performance

The prototype presented in Chapter 3 serves primarily as evidence of a practically implementable chaotic communication system; it also serves as an tangible introduction to the specific needs for robust chaotic circuit/waveform acquisition and modified signal processing techniques generalized from DS spread spectrum communications. This chapter provides a general foundation for practically implementable chaotic communication systems, exploring signal acquisition, synchronization mechanisms, and signal processing techniques, including measured data from the prototype chaotic communication system for evaluation and comparison to the analytical predictions. Subsequent chapters then discuss specific extensions of the core chaotic communications techniques presented in this chapter, generalizing the basic chaotic waveform to a potentially new class of maximal entropy waveforms, each with specific enhancements.

4.1 Chaotic Waveform Acquisition and Synchronization

The chaotic waveform presents a unique challenge in terms of performing robust acquisition under practical channel conditions. The signal has a non-stationary amplitude, providing potential problems to traditional matched filter approaches. Moreover, the impulsive autocorrelation of the waveform requires that the signal be captured precisely in a narrow acquisition time window in order to make an immediate decision. This may be mitigated by taking extremely long correlations, which are in turn impacted by frequency offsets, or by inserting multiple ambles at predefined intervals for the receiver to lock onto. An innovative approach, termed an adaptive correlator[156], flexibly addresses these various constraints and provides a robust acquisition method with iterative Bayesian correlation results to provide an initial chaotic state lock comparable to Pecora's generalized synchronization[5, 43]. Provided the various time drift and frequency offsets of the received waveform are within the loop bandwidths of the tracking filters, facilitated by transmission of a known data preamble, the correlator provides something closer to the asymptotically convergent identical synchronization (aka Pecora-Carroll synchronization[32]).

This section presents the system-level receiver acquisition state machine, a brief analysis of the required acquisition time window, and presentation of the adaptive correlation techniques. Specific adaptive correlator topics discussed include an analytical derivation of correlator thresholding, decision metrics, variations in correlator adaptation, hardware implementation, signal offset estimation, and comparative performance to traditional correlation

methods. The end result of the adaptive correlator developed in this section is a flexible and efficient acquisition engine that robustly locks onto the chaotic waveform in approximately 10-20% of the time required by traditional methods at low SNRs.

4.1.1 Chaotic Receiver Acquisition State Machine

The general approach to chaotic waveform acquisition models that of a direct sequence spread spectrum system: complete internal hardware synchronization (e.g. clock via DCM to external reference), perform a coarse time synchronization of the internal sequence generators to the expected state of the received signal, and then perform a correlation-based acquisition scheme to determine actual offsets. Additionally, some method of handling acquisition failures at the receiver is required so that the receiver may re-attempt acquisition. In the prototype communication system, a single preamble of 200 known signals is used for initial acquisition and tracking, with knowledge at the receiver that the transmitter will transmit on a GPS-referenced 1PPS pulse. The receiver repeats its attempts at acquisition on each and every 1PPS pulse until it achieves a lock; higher level network protocol functionality governs the handling of acquisition timeouts. A summary timeline of physical layer acquisition processing of the chaotic waveform transmission is shown in Figure 66.

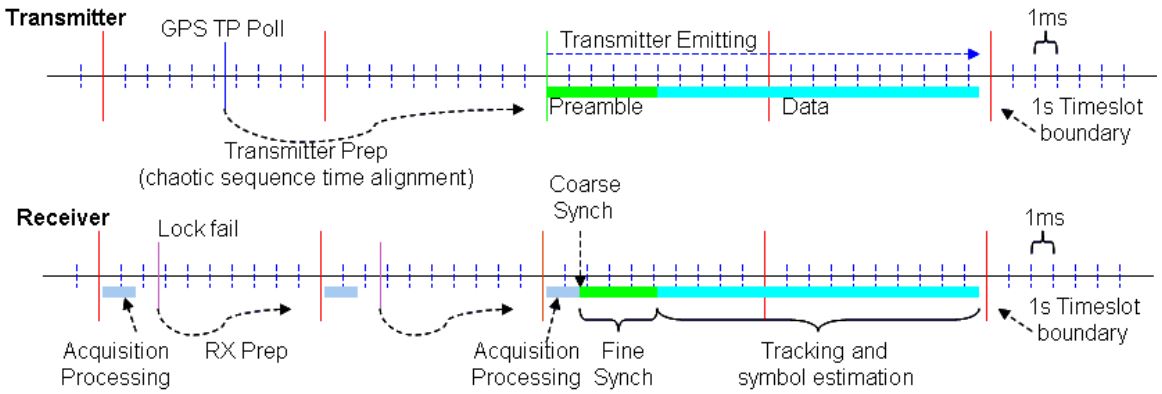

FIGURE 66. Coherent chaotic receiver acquisition processing.

4.1.2 Chaotic Receiver Acquisition Window

Determining the necessary acquisition window requires aggregation of all timing uncertainties that exist between the two physical platforms. The timing uncertainty in the signal is caused by the accuracy of the oscillator and the time elapsed since the last GPS pulse. Assuming a crude 100 ppm error on the internal oscillator and a 100 μs interval between GPS pulse and conclusion of the acquisition window, the clock has an uncertainty of

$$(100\,ppm) \cdot (100\,\mu s) = 10\,ns$$

This time uncertainty occurs on both the transmitter and the receiver, making the net timing uncertainty ±20 ns. Dominant to this is the timing uncertainty of the GPS references, which are on the order of 1 μs of timing uncertainty for the 1PPS timepulse. In addition, the receiver must account for the propagation delay of the transmitted signal; assuming a range of up to 1000 m for rooftop testing, the propagation delay is between 0 and 3.3 μs. The chosen time window of 6.4 μs centers the expected arrival of the transmitted chaotic waveform with a small coverage of additional uncertainties caused by RF circuitry (measured to be static at $\approx$ 20 ns), and also yields a convenient search over 256 relative delays at 40 MHz baseband sample rate. A timing diagram of the uncertainty window is shown in Figure 67.

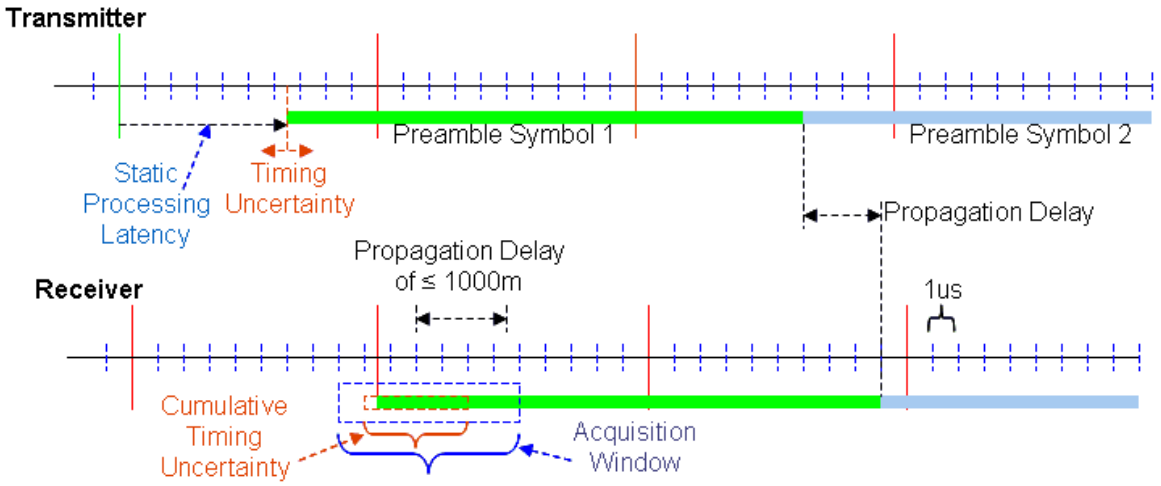

FIGURE 67. Coherent chaotic receiver acquisition window.

4.1.3 Adaptive Correlation Techniques

The heart of the chaotic waveform acquisition method is the adaptive correlator[157], which is a Bayesian time-domain correlation estimator: iterative correlations of increasing size are used to predict whether the chosen correlation time window contains the correlation peak. This brief time window is then stepped through a period of time equal to the acquisition window until a lock is found or failure is declared. A strong preference towards false accepts is encoded in shorter-length correlation metrics to ensure that intermediate correlations resulting in a negative decision are unlikely to contain the desired correlation peak. Moreover, the state machine defining the correlator processing searches over overlapping frequency offsets to better ensure a positive lock. Once a signal proceeds through two intermediate correlations successfully, a relatively long correlation is performed to ensure that the declared lock is derived from the desired signal. A final correlation is performed to gather tracking loop initialization values before the adaptive correlator shifts from acquisition mode to demodulation mode. A block diagram of an exemplary adaptive correlator is shown in Figure 68.

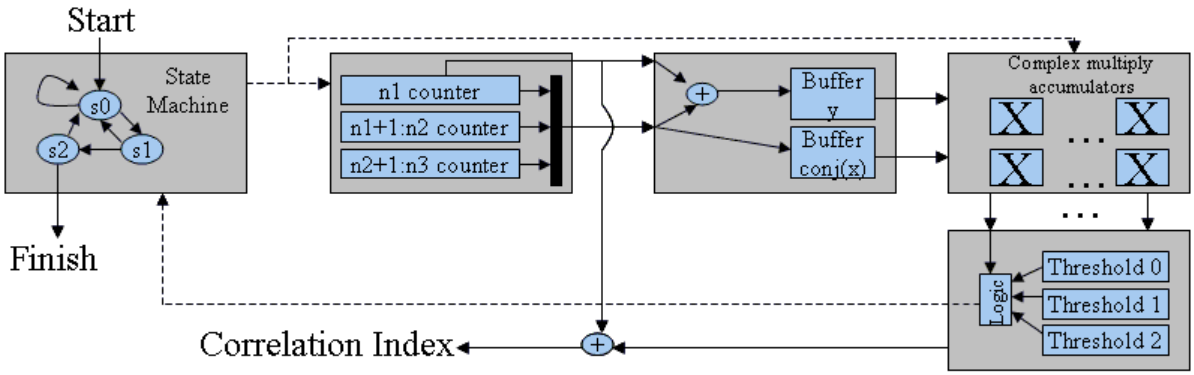

FIGURE 68. Block diagram of an adaptive correlator.

Throughout this section, an analytical framework for adaptive correlation techniques will be presented, starting with the high-level design considerations and methods of processing intermediate correlation data leading to an exemplary hardware implementation. Correlation result estimators will be presented for quantifying the time and frequency offsets from the final correlation outputs; thees estimates provide more robust tracking loop initialization by jamming initial values. A third estimator for phase rotation was initially developed, but later moved to a post acquisition stage since the phase rotation of a large frequency offset during the acquisition processing results in a practical uniformly random phase offset. Finally, a summary of simulated adaptive correlation results is provided and validated by prototype hardware measurements.

4.1.3.1 Adaptive Correlator Design

The adaptive correlator relies on a controlled series of time domain correlations to winnow the time-frequency search span required to obtain a high reliability correlation lock. Generally, the goal of a time domain correlation based acquisition processor is to obtain the relative delay $k_\star$ corresponding to the maximum correlation magnitude and then determine whether that peak is sufficiently high to declare lock. For completeness of discussion, let X and Y be baseband samples from the internally generated chaotic sequence and the digitized receive chaotic waveform, respectively.

$$X = \{x_1, x_2, \ldots, x_k, \ldots x_L\} \qquad Y = \{y_1, y_2, \ldots, y_k, \ldots y_L\}$$

A length-R correlation between the two sequences having relative delay k_0 is calculated as

$$\sum_{k=\hat{k}}^{R+\hat{k}-1} y_k x^*_{k-k_0}$$

Note that this cross-correlation is not necessarily time invariant when provided a nonstationary input, so different choices of $\hat{k}$ will result in different cross-correlation values, even for the same value of k_0. Regardless, the processing required to obtain the length-R correlation value is R complex multiplications and $(R-1)$ complex additions. To perform length-R correlations across the entire acquisition window of K potential relative delays $\hat{k}$[36] and H potential frequency offsets requires RKH complex multiplications. The fundamental goal of the adaptive correlator is to reduce this number of operations, re-using hardware where possible, yet return a similar assurance of correlation peak lock or failure.

Returning to the content of sample sequences X and Y, the two sequences may be simplified by recognizing the impulsive autocorrelation of the chaotic waveform. At the optimal relative delay $k_0 = k_\star$, the received signal Y is identical to X within a complex-valued scaling factor α.

$$y_k = \alpha x_{k-k_\star} = |\alpha| e^{j\phi(\alpha)} x_{k-k_\star}$$

In addition, the received signal contains an additive noise component that may be written as a similar sequence N and is assumed to have a uncorrelated Gaussian noise characteristic. Therefore, the more complete evaluation for a length-R correlation is

$$\begin{aligned}\sum_{k=\hat{k}}^{R+\hat{k}-1} x^*_{k-k_0} \cdot (y_k + n_k) &= \sum_{k=\hat{k}}^{R+\hat{k}-1} x^*_{k-k_0} \cdot \left(|\alpha| e^{j\phi(\alpha)} x_{k-k_\star} + n_k \right) \\ &= |\alpha| e^{j\phi(\alpha)} \sum_{k=\hat{k}}^{R+\hat{k}-1} x^*_{k-k_0} x_{k-k_\star} + \sum_{k=\hat{k}}^{R+\hat{k}-1} x^*_{k-k_0} n_k\end{aligned}$$

The first summation in the correlation result is effectively an integrated signal power during the length-R time window, while the second correlation is a random term that accounts for the background noise in the received signal. Note that the relative amplitudes of X and N may be chosen arbitrarily by the use of an AGC loop and binary representation. As such, subsequent calculations will assume that the power of sample sequence X, σ_x^2, and the power of the noise signal N, σ_n^2, are identically equal to unity, which is equivalent to mapping both the received noise and the internally generated chaotic sequence to standard Normal distributions. As a result, the scaling term α represents both an attenuation $|\alpha|$ and a static phase offset $e^{j\phi(\alpha)}$ under the assumption of zero phase drift (frequency offset) during the captured R correlation points;[37] simple bounds on the frequency offset (± 7.5 kHz, corresponding to a drift of ≈ 0.9

[36] Note that if the sequences X and Y are sampled in an non-decimated fashion, then $L \geq R + K$.

[37] The primary goal of using the chaotic waveform in a maximal entropy communication system is to minimize the transmitted energy, thus indicating that the received spread waveform will be well below the noise floor ($|\alpha| << 1$) and then recovered via coding gain. As such, the receiver AGC function operates strictly on the background noise level and does not make an appreciable bump when a bursted chaotic signal appears. An alternate solution is to burst the initial portion of the acquisition preamble (approx 6 μs) at a higher power level than the remainder of the preamble or modulated data; randomizing the preamble start time will improve multiple access communications performance by staggering bursts.

radians during a 768-point correlation at a 40 MHz sample rate) ensure that the phase drift during the correlation does not destroy the detection probabilities.

The adaptive correlator block diagram described previously implements a three stage Bayesian detection estimation, starting at sample $k = 1$ and performing a coarse-grade correlation of $R = N_1 = 32$ points, a medium-grade correlation of $R = N_2 = 224$ points, and a fine-grade correlation of $R = N_3 = 768$ points; all correlations are performed on disjoint subsets of k, leading to largely independent results. The choice of R for each of these iterative correlation steps depends on the expected carrier-to-noise ratio, which may be derived directly from $|\alpha|$. The state machine controls the portions of the two sequences that are compared ($\hat{k}$) as well as the relative delay between the sequences (k_0). After performing each coarse-grade N_1-point correlation, a decision is made to either advance the relative delay, increment the potential frequency offset setting, or perform a medium-grade N_2-point correlation on the same sequence subset. The thresholding that decides the subsequent state from the correlation value is also a strong function of the expected $|\alpha|$. A general state flow diagram for a three-stage adaptive correlator is shown in Figure 69; this design concept may be extended arbitrarily to different length correlations, different hardware allocations, correlations over non-contiguous input sequences buffered in memory (selective noise cancellation approach discussed in later sections), or based on decimated subsets of k.

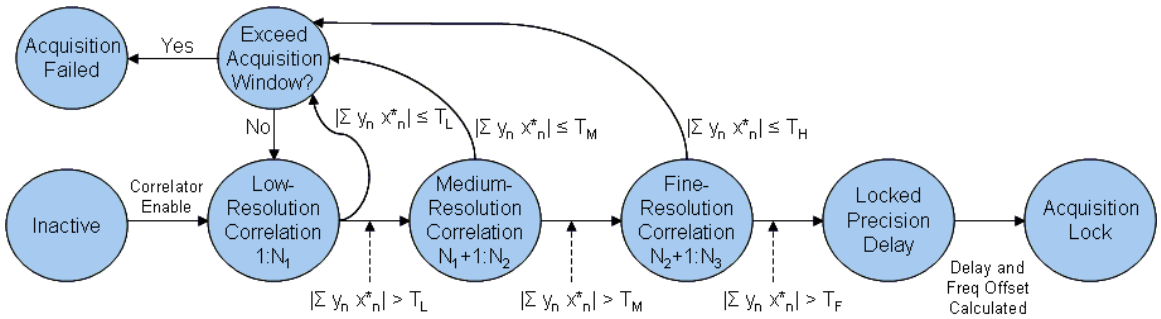

FIGURE 69. State machine of an adaptive correlator.

For completeness, the adaptive correlator must search across the valid range of frequencies and potential relative delays. Adding in an induced frequency offset term $\hat{f}$, the actual correlator result becomes

$$\sum_{k=\hat{k}}^{R+\hat{k}-1} e^{j2\pi\hat{f}/f_s(k-\hat{k})} x^*_{k-k_0} \cdot (y_k + n_k) = \sum_{k=\hat{k}}^{R+\hat{k}-1} e^{j2\pi\hat{f}/f_s(k-\hat{k})} x^*_{k-k_0} \cdot \left(|\alpha| e^{j\phi(\alpha)} x_{k-k_\star} + n_k\right)$$

$$= |\alpha| e^{j\phi(\alpha)} \sum_{k=\hat{k}}^{R+\hat{k}-1} e^{j2\pi\hat{f}/f_s(k-\hat{k})} x^*_{k-k_0} x_{k-k_\star} + \sum_{k=\hat{k}}^{R+\hat{k}-1} e^{j2\pi\hat{f}/f_s(k-\hat{k})} x^*_{k-k_0} n_k$$

When a frequency offset exists in the input signal, the previously static α scaling term may be modeled as having a time-varying phase; when the frequency offset is too large, the

previously coherent summation (viewed as a vector summation in $\mathbb{C}$) will begin to rotate during the correlation window and cancel itself out. A depiction of the intermediate correlation accumulations over 1024 points and a frequency offset of $\Delta_{freq} = 7.5$ kHz is shown in Figure 70. The ideal correlation vector lies strictly on a ray from the origin at $\phi(\alpha)$.

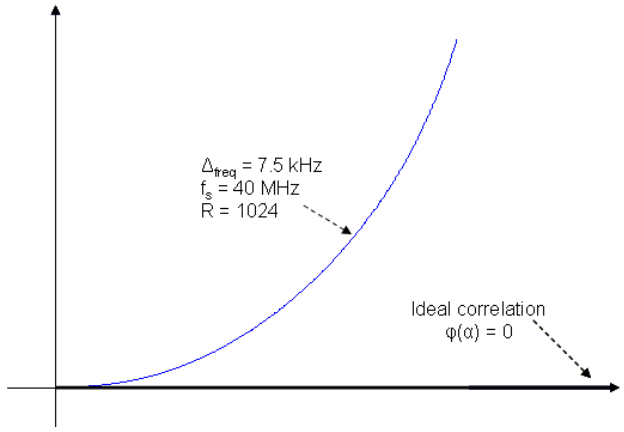

FIGURE 70. Correlation accumulations with 7.5 kHz frequency offset.

The effective loss in signal energy for a 7.5 kHz offset is only 5% or 0.2 dB, while the loss for a 15 kHz offset is 0.9 dB and for 30 kHz offset is closer to 5.5 dB. As a result, the adaptive correlator has a coarse internal sine/cosine generator to mimic a frequency offset during the correlation, attempting to negate $\hat{f}$; a total of five settings, $\{0, \pm 15, \pm 30\}$ kHz, were included, producing a sawtooth phase pattern during intermediate correlations (cosine in blue and sine in green) as shown in Figure 71.

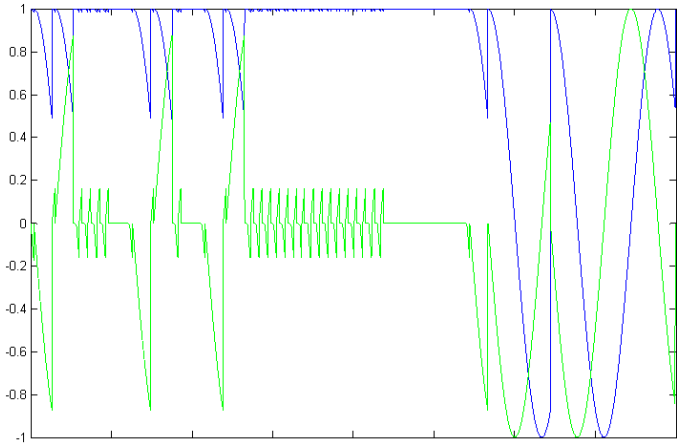

FIGURE 71. Phase adjustments in intermediate adaptive correlation accumulations.

Viewing the adaptive correlator state diagram as a finite state Markov chain, the goal is to minimize the probabilities of moving forward in the chain unless the desired signal is located. The relative efficiency of the adaptive correlator can even be measured as the sum

of all multiplication operations required compared to that of a fixed correlator design at the maximum correlation length prior to a lock decision. More precisely, the state machine may be reduced to a set of asynchronous states (temporarily ignoring the acquisition fail state, which occurs after numerous relative delays are processed without locking on a desired signal).

State 1:	Inactive
State 2:	Low-Resolution Correlation
State 3:	Medium-Resolution Correlation
State 4:	Fine-Resolution Correlation
State 5:	Locked Precision Delay
State 6:	Acquisition Lock

Each state has transition probabilities $P_{i,j}$ of moving from State i to State j at the next decision. Since only a limited number of these states transitions are valid, the transition matrix collapses to $[T_{i,j}]$.

$$T_{i,j} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & P_{2,2} & P_{2,3} & 0 & 0 & 0 \\ 0 & P_{3,2} & 0 & P_{3,4} & 0 & 0 \\ 0 & P_{4,2} & 0 & 0 & P_{4,5} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

As written, all row sums ($\sum_j P_{i,j}$) will be equal to unity. Ideally, thresholds are chosen such that $P_{2,2} > P_{2,3}$, $P_{3,2} > P_{3,4}$, and $P_{4,2} > P_{4,5}$ for all but the desired signal delay (the value of $P_{4,5}$ is the probability that the correlator will lock on the wrong signal during a fine-resolution correlation). For the certainty provided by a length-N_3 correlation (which is approximately $1-P_{4,5}$), the adaptive correlator requires an expected $P_{2,2}N_1+P_{2,3}(N_2-N_1)+P_{2,3}P_{3,4}(N_3-N_2)$ multiplications per delay to throw out the potential delay as a candidate solution. Note that the efficiency savings

$$\eta = 1 - \frac{P_{2,2}N_1 + P_{2,3}(N_2 - N_1) + P_{2,3}P_{3,4}(N_3 - N_2)}{N_3 - N_2} = 1 - P_{2,3}P_{3,4} - P_{2,2}\frac{N_1}{N_3 - N_2} - P_{2,3}\frac{N_2}{N_3 - N_2}$$

can be negative if poor thresholding decisions are used. Further, extending the size of the adaptive correlator state machine to more than a 3-stage coarse, medium, and fine correlation does not necessarily give more efficient operation. As a numerical example, consider an adaptive correlator with the following parameters

$$N_1 = 32 \qquad N_2 = 256 \qquad N_3 = 1024 \qquad P_{2,2} = P_{2,3} = 0.5 \qquad P_{3,4} = 0.05 \qquad P_{4,5} = 0.0005$$

that results in a 21% lower processing for nearly the same statistical assurance as a consistent 768-point correlation.

$$\eta = 1 - 0.5 \cdot 0.05 - 0.5 \cdot \frac{32}{768} - 0.5 \cdot \frac{256}{768} = 78.75\%$$

The probabilities of false accept and false reject must be evaluated relative to the carrier to noise ratio during operations, with increased R values at each correlation step; further, some of the saved processing time is used for new operations such as clearing the CMACs and generating intermediate decision metrics. In general, the acquisition processing of an adaptive correlator reduces linearly with the number of CMAC cells (parallelized multiplications) and marginally by fully pipelining the correlation and decision process. Emperical results have shown that a three- to four-state adaptive correlator is ideal, with more states used at lower expected carrier-to-noise levels. Further, the adaptive correlator design may be improved by decimating the baseband samples to a rate more comparable with the spread bandwidth. The two diagrams shown in Figure 72 demonstrate simulated correlator output values during a successful correlation with $\alpha = 0.1$ (left) and the measured response through the adaptive correlator states[38] and intermediate correlation values during a correlation with $\alpha = 1$ (right); note that the hardware model for the adaptive correlator clears the output value once the hardware block is disabled.

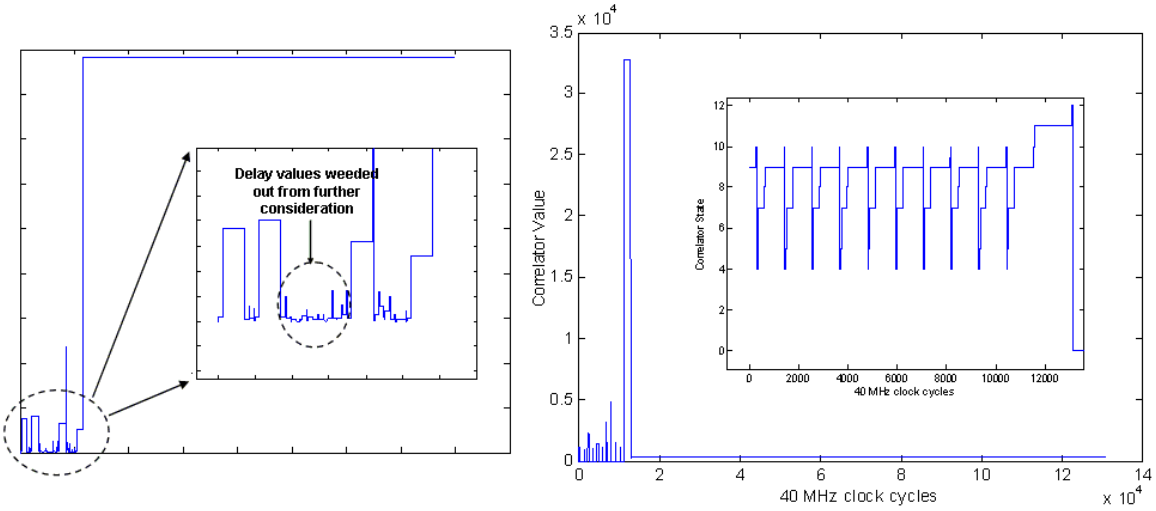

FIGURE 72. Measured outputs of a hardware adaptive correlator, demonstrating intermediate correlation values (left) and state adaptation during a failed correlation (right).

4.1.3.2 Adaptive Correlator Thresholds

The most difficult part of using the adaptive correlator is choosing the proper threshold values. Choose them too small, and the efficiency gains of adaptability are negated. Choose them too large, and the likelihood of triggering on the desired signal is reduced. As described previously, the correlation result for any length-R correlation is

$$\sum_{k=\hat{k}}^{R+\hat{k}-1} x^*_{k-k_0} \cdot (y_k + n_k) = \sum_{k=\hat{k}}^{R+\hat{k}-1} x^*_{k-k_0} \cdot \left(|\alpha| e^{j\phi(\alpha)} x_{k-k_\star} + n_k \right) = |\alpha| e^{j\phi(\alpha)} \sum_{k=\hat{k}}^{R+\hat{k}-1} x^*_{k-k_0} x_{k-k_\star} + \sum_{k=\hat{k}}^{R+\hat{k}-1} x^*_{k-k_0} n_k$$

[38] A detailed Matlab script implementing the control structure for an exemplary adaptive correlator, including reference state numbers is contained in Appendix A.

When the desired signal is present within the correlation time window ($k_0 = k_\star$), the first summation will be a coherent sum with a nearly Gaussian statistical distribution and the second summation will be a sufficiently Gaussian statistical distribution ($R > 12$ assumed). The statistical distributions[39] have the probability density functions:

$$f_1(u) = \frac{u^{\frac{R}{2}-1}e^{-\frac{u}{2}}}{2^{\frac{u}{2}}\Gamma(\frac{u}{2})} = \frac{u^{\frac{R}{2}-1}e^{-\frac{u}{2}}}{2^{\frac{u}{2}}\left(\frac{u}{2}\right)!} \approx \frac{1}{2\sqrt{\pi R}}e^{-\frac{(u-R)^2}{4R}} \qquad f_2(u) = \frac{1}{\sqrt{2\pi R}}e^{-\frac{u^2}{2R}}$$

The first distribution converges to a constant-mean Gaussian distribution for large R via the Central Limit Theorem, $N(|\alpha|R, \sqrt{|\alpha|\sqrt{2R}}^2)$, yet has a median that is slightly less than the mean. The second distribution converges to a zero mean Gaussian distribution with variance R, $N(0, \sqrt{R}^2)$. Since the correlator thresholds are chosen based on total energy, complex-valued phase, $\phi(\alpha)$ does not play into the decision. When the desired signal is not present in the current time window ($k_0 \neq k_\star$), the first summation is equal to the correlation of un-correlated samples (outside ± 3 speading chips), with a similar distribution to that of the background noise, converging to a $N(0, (|\alpha|\sqrt{R})^2)$ distribution. Simplifying these approximated distributions to distinct random variables, the three distinct correlation contributors are

$$V_1 \equiv N(|\alpha|R, (|\alpha|\sqrt{2R})^2) \qquad V_2 \equiv N(0, \sqrt{R}^2) \qquad V_3 \equiv N(0, (|\alpha|\sqrt{R})^2)$$

The sum of two Normal random variables is again Normal, leading to the following expected distributions.

Signal Present: $V_1 + V_2 \approx N(|\alpha|R, \sqrt{(1+2|\alpha|^2)R}^2)$ Signal Absent: $V_2 + V_3 \approx N(0, \sqrt{(1+|\alpha|^2)R}^2)$

The correlation threshold must be chosen sufficiently large that the combination of signal variance and noise contribution do not cause a false decision. For a given threshold T, the probabilities of true/false accept/reject, ignoring the negative tail of the signal present case, are as follows.

$$\begin{aligned} P(\text{True Accept}) &= P(V_1 + V_2 \geq T) \\ P(\text{False Accept}) &= P(|V_2 + V_3| \geq T) = 2P(V_2 + V_3 \geq T) \\ P(\text{True Reject}) &= P(|V_2 + V_3| < T) \\ P(\text{False Reject}) &= P(V_1 + V_2 < T) \end{aligned}$$

As an example, for a signal with 0 dB spread carrier-to-noise ratio ($|\alpha = 1|$), and a correlation size of $R = 64$, choosing the threshold $T = 16$ for the absolute value of the correlation provides a false accept rate of 15.7% and a false reject rate of 0.023%. Simulated

[39] Without loss of generality, R is assumed even for reduction of the Gamma function.

values for 16M samples in Matlab show a similar result at 15.5% and 0.00%, respectively. Moreover, the variance of the desired signal is effectively less since the skewness and excess kurtosis are both significantly greater than zero, indicating a shift in mass towards the right distribution tail; treating the high-order Chi-squared distribution as a Gaussian distribution is a worst-case estimate. In general, the choices of adaptive correlator thresholds should be made with sufficient values to ensure that the known signal passes while reducing unnecessary processing in subsequent states. A second example is a two-state adaptive correlation with $N_1 = 64$ points, $N_2 = 256$ points and a -6 dB carrier-to-noise ratio for the spread bandwidth ($|\alpha| = 0.5$). Choosing the first threshold as 8 results in a false accept rate of 37.1% and a false reject rate of 0.73%, validated as 36.7% and 0.34% in Matlab; choosing the second threshold as 64 results in a false accept rate of 0.026% and a false reject rate of 0.052%, validated as 0.027% and 0.038% in Matlab. This second example cuts the average number of multiplications per correlation from 256 to 134.5, which is a 47% savings.

4.1.3.3 Adaptive Correlator Decision Metrics

Along with the decision of threshold levels, there remains a decision as to the how the correlator results are compared to the threshold when aggregated for parallel processing. The primary consideration is a bank of N_{ma} CMACs, each of which processes time-domain correlations for an independent relative delay k_0; when N_{ma} is large, the aggregation of correlation results into the threshold decision metrics tends towards more complex order statistics rather than straightforward Bayesian estimations. In addition, the estimates developed to this point do not account for the oversampling of the baseband samples or the finite width of the correlation peak – both of these considerations play into the threshold decisions.

4.1.3.3.1 Simple Adaptive Correlator Decision Metric

The simplest decision metric for the adaptive correlator is to make a decision on all delays simultaneously, pursuing further processing of the N_{ma}-baseband sample time window as a fixed grouping. This method was employed in the prototype chaotic receiver by comparing the maximum of the eight multiply accumulator magnitudes to a chosen threshold. The actual value of the correlation and the corresponding delay index k_0 are passed through a magnitude comparison tree as well for future manipulation of the intra-correlation phase rotation to derive an initial frequency offset estimate in the tracking loops.

This approach simplifies the hardware implementation and control logic by considering each set of inputs as a block, but it makes the mathematics underlying the thresholds more difficult (must consider order statistics) and is less efficient since an entire block is processed into the next correlation tier even if only one potential delay had a value exceeding the threshold. Since the correlation peak has an approximate width of two 10 MHz samples, equal to eight 40 MHz samples, the processing of a time window of eight samples, each offset by relative delays of $\frac{1}{40MHz} = 25$ ns, is nearly equivalent to search over two distinct chaotic sequence samples. This implies the need to consider a modified order statistics approach with the maximum of two independent inputs instead of eight.

In the present application, the correlator efficiency is not an absolute requirement, but it is beneficial in that higher efficiency correlation shortens the transmission preamble; as the number of parallel multiply accumulators increases, the *maximum* operator becomes even more

inefficient. Practical values of N_{ma} in this configuration are probably limited by $\approx 15 - 20$.[40]

4.1.3.3.2 Efficiency Optimized Adaptive Correlator Decision Metric

To obtain the maximum efficiency from the correlator, it would be beneficial to permit each of the parallel multiply accumulator structures to be under independent control. This idea can be taken as far as creating entirely independent state machines, where the indices being searched are controlled separately (the clock frequency is identical in all cases, but the threshold decisions and state machines evolve independently) and can share the pool of potential delays k_0. In this fashion, the first N_{ma} potential delays will be searched similarly to the first approach after first enabling the correlator. The decisions on the outputs will be handled independently (either the threshold may be tightened or else increase computational efficiency). As an example, consider a case where the fifth of ten correlator magnitudes exceeds the chosen first threshold for a 50-point correlation ($k = 5$, $N_{ma} = 10$, and $N_1 = 50$). The fifth multiply accumulator will trigger itself to advance to the next correlation using indice $k + N_1 + 1 = 56$ and forward for the Y sequence and indice $N_1 + 1 = 51$ for the X sequence; this maintains the relative delay of $k = 5$, but advances the indices to correlate on an independent sample set. The other $N_{ma} - 1 = 9$ multiply accumulators progress onto the next $N_{ma} - 1$ potential delays, starting at $N_{ma} + 1 = 11$. Thus in the second round of correlations, potential delays of $\{5, 11, 12, 13, 14, 15, 16, 17, 18, 19\}$ samples are being investigated. The difficulty occurs when $N_2 \neq mN_1,\ m \in \mathbb{N}$,[41] and therefore the parallel set of multiply accumulators becomes de-synchronized from a control perspective. One option is to simply delay the advanced searches for any control periods of the others, or else allow the multiply accumulator cells to have completely independent control. A global control mechanism must be maintained to instruct each multiply accumulator which potential delays to investigate next when returning to the lowest resolution correlation.

Overall, the hardware trade makes this both the most efficient in search time and in hardware; the added control logic will be significantly less than that required to construct additional multipliers for matching the same efficiency gains. Further, each of the individual control logics will be similar, allowing design re-use.

[40] Another variant of this method is to make a decision on the sum of the magnitudes rather than the maximum. With oversampling at the receiver, the correlation peak has a finite width, yet will not necessarily be centered in the intermediate correlation window. Likewise, a complex-valued vector addition may be done prior to the magnitude calculation and threshold comparison, yet the noise will also be correlated between successive sample durations.

[41] Adjusted for latencies induced by the control structure.

4.1.3.3.3 Improved Adaptive Correlator Searches

Two additional methods that will be useful for improving adaptive correlator efficiency are a smart-search algorithm that begins the search around previously locked conditions (e.g. for signal re-acquisition, periodic ambles, or burst communications) and a variable decimation approach that first performs correlations at relative delays comparable with the spread bandwidth sample rate, expanding the search to contiguous fine-grained delays when an intermediate correlation exceeds the chosen threshold. In both cases, the control structure becomes more complex, yet requires minimal amounts of additional hardware. A final potential improvement to the adaptive correlator design is to harness the selective noise cancellation technique[158] discussed in Section 4.2.1.3 for acquisition processing at negative spread carrier-to-noise ratios.

4.1.3.3.4 Hybrid Adaptive Correlator Decision Metric

A method that merges the simpler and the more efficient approaches discussed previously is to follow through the threshold comparison decisions with a maximum correlation value as in the simple case, yet accelerating the indices as much as possible when the initial thresholds are exceeded. For example, consider a correlation that results in the eighth of ten parallel multiply accumulators exceeding the threshold (only the eighth exceeds); the subsequent step that pursues the next tier correlation will start with delay 8 rather than delay 1 from the previous case. As a result, an additional 7 potential delays are measured (at the higher resolution) for free. If the trigger is a false alarm due to random noise, ignoring the case where two noise sequences independently exceed the threshold, then the expectation is that $\frac{N_{ma}}{2}$ delays are skipped each time a false alarm is registered. Therefore, some increase in efficiency is obtained with almost no hardware or control logic addition.

4.1.3.4 Adaptive Correlator Implementation

Unlike direct sequence spread spectrum, the multiplications used to receive a chaotic waveform cannot be reduced to simple accumulation that switches between addition or subtraction – all multiplication operations must be done with hardware multipliers or multi-bit accumulators. At one end of the implementation scale is a fixed pipelined structure of R complex multipliers (similar to an adaptive equializer) that permits near real-time decisions, while at the other end is a serial processor that performs all arithmetic as shift-add operations on a stored buffer. The improved efficiency of the adaptive correlator search process can be translated into some combination of reduced hardware or acquisition latency. Given the one second duration between acquisition attempts and the likely preference for a portable hardware

platform in practical communication applications, the chosen method for the hardware prototype coherent chaotic communication system tends towards the middle of the range, using a static acquisition buffer and eight CMAC cells. This section covers the implementation of the adaptive correlator state machine for search across the span of frequencies and relative delays, acquisition buffer indexing structure, efficient three-multiplier CMAC cells, correlator threshold and decision, and finally a hardware utilization summary for the physical prototype. A top-level diagram of the adaptive correlator Synplify DSP implementation is shown in Figure 73.

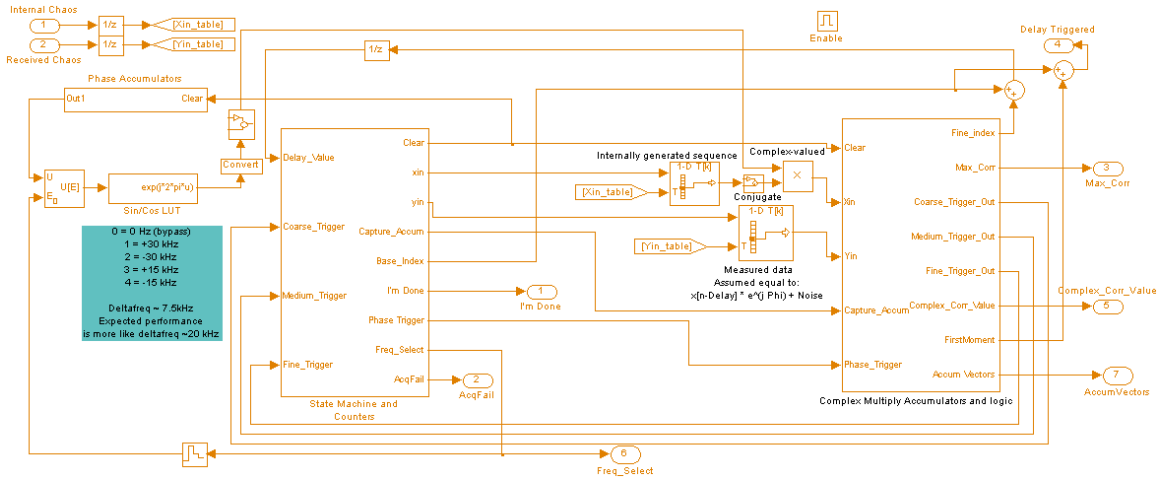

FIGURE 73. Synplify DSP adaptive correlator implementation.

4.1.3.4.1 Adaptive Correlator State Machine

Implementation of the adaptive correlator state machine is relatively trivial, with three trigger lines indicating the current state – once the adaptive correlator exceeds the chosen threshold for the coarse correlation, the *Coarse Trigger* forces the state machine into running a medium correlation. The state machine was designed using asynchronous, externally enabled logic to control correlation length and threshold selection for the next intermediate correlation. The state machine also provides an incremented time index *Base Index* corresponding to the current relative delay k_0 and an incremented frequency offset index *FreqSelect* that commands the current induced frequency offset. After declaring a correlation lock, the state machine performs a final correlation, centered at the expected relative delay $k_0 = k_\star$, collecting 3 intermediate correlation results for subsequent frequency offset estimation. During implementation, the correlator state machine was hard coded in a M-control file to ensure proper behavior (script located in Appendix).

4.1.3.4.2 Adaptive Correlator Sequence Indexing

The sequence indexing structure was constructed using selectable counters from $\{1, 2, \ldots, N_1\}$, $\{N_1+1, N_1+2, \ldots, N_2\}$, and $\{N_2+1, N_2+2, \ldots, N_3\}$ that are fed into a length-2048 acquisition buffer. The y_{in} index is adjusted by adding the array $[0 : (N_{ma} - 1)]$ to generate a distinct time window of received signals, while the x_{in} index is adjusted to maintain the current relative delay k_0 between sequences. As a result, only one complex multiplication is required to induce the frequency offset in the correlations, and the y_{in} index may be implemented as a N_{ma}-block memory access. During the final correlation, a backstep of size $\frac{N_{ma}}{2} - 1$ is induced to the expected correlation peak index $\hat{k_\star}$ obtained during lock declaration in order to center the correlation peak in the N_{ma} correlation window.[42]

4.1.3.4.3 Adaptive Correlator CMACs

The core computational engine of the adaptive correlator is the complex multiply accumulator (CMAC) banks, providing controlled correlation processing based on the state machine commands. The general structure of the CMAC bank is depicted in Figure 74.

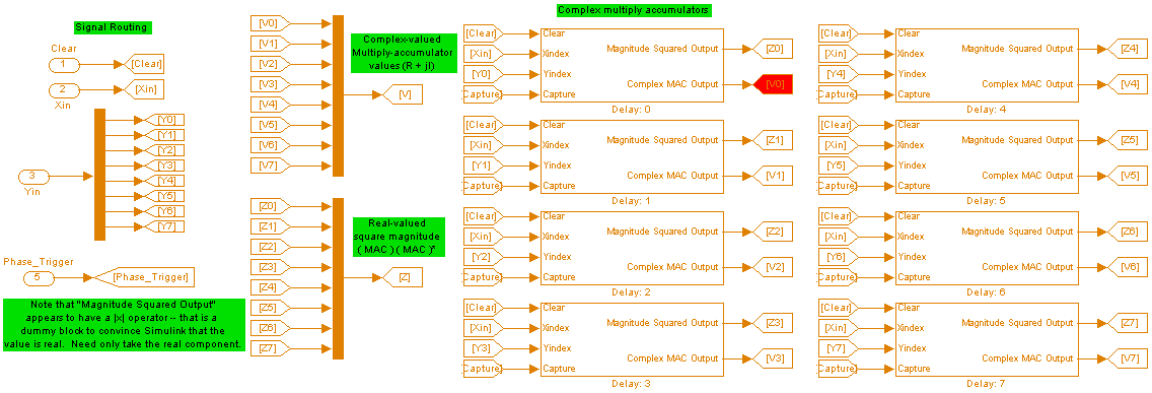

FIGURE 74. Simulink CMAC bank.

The control structure for the CMACs implements a traditional integrate-and-dump sequencing based on the *Clear* input. The x_{in} input is the frequency-offset adjusted sample of the internally generated chaotic sequence at index $k = k_0 + \hat{k}$. The y_{in} input is actually an array of N_{ma} successive values of the received signal at indices $k = \{\hat{k}, \ldots, \hat{k} + N_{ma} - 1\}$. The outputs of the CMACs, $\vec{Z}$ and $\vec{V}$ are the squared complex magnitude and complex value of the N_{ma} successive correlations, respectively. $\vec{Z}$ and $\vec{V}$ are used by the subsequent decision metrics to determine whether the accumulated values exceed the threshold. The structure of the CMAC is made slightly more efficient by using a three-multiplier inner product topology

[42] A further improvement may be implemented for large correlations by replacing the coarse frequency step $\hat{f}$ with an estimated value.

by recognizing,

$$(a + jb) \cdot (c + jd) = (a \cdot c - b \cdot d) + j\left((a + b) \cdot (c + d) - a \cdot c - b \cdot d\right)$$

resulting in an exemplary topology shown in Figure 75.

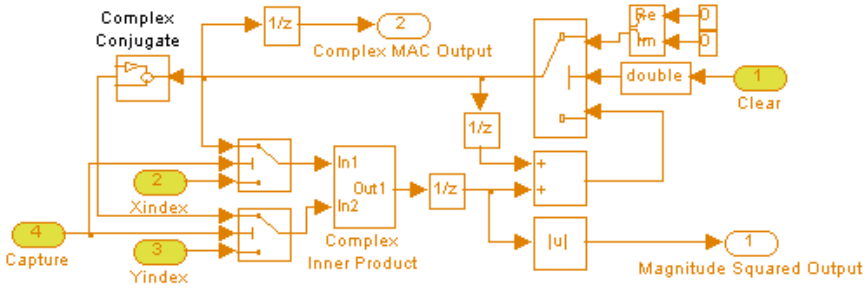

FIGURE 75. Simulink CMAC implementation.

During the final correlation cycle, the *Capture* pulse re-directs the accumulated value and its conjugate to the input of the complex inner product to perform a complex magnitude computation; an additional line was added during the final hardware implementation to the scale value within the inner product, retaining only the relevant MSBs before decision metrics are applied.

4.1.3.4.4 Adaptive Correlator Thresholding and Decision Metrics

At the conclusion of each correlation the complex value $\vec{V}$ and magnitude $\vec{Z}$ are passed to the thresholding block shown in Figure 76.

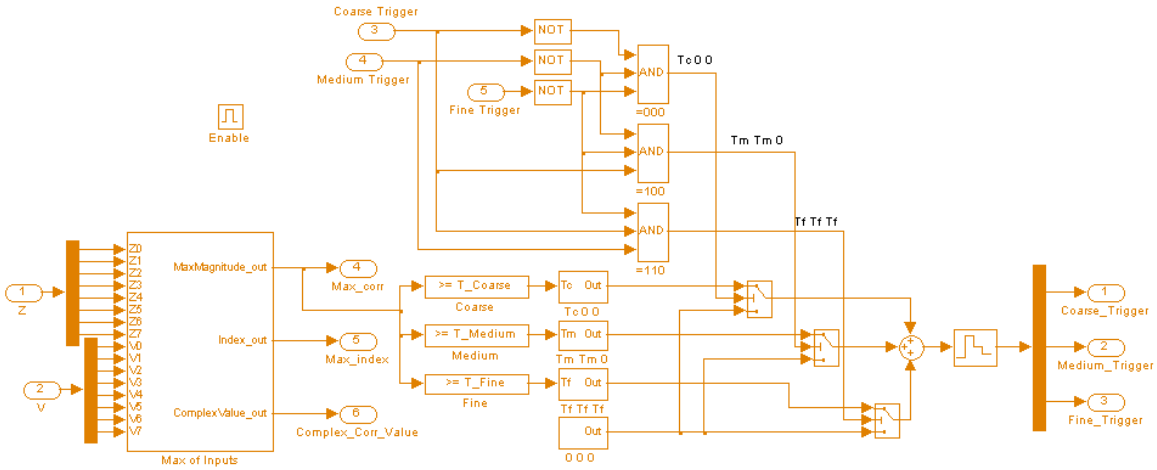

FIGURE 76. Adaptive correlator thresholding and decision metrics.

First, the complex magnitudes are compared for selection of the maximum magnitude, corresponding complex value, and CMAC index (integer 0 to 7). This maximum magnitude is

then compared to each of the coarse, medium, and fine thresholds for determination of whether to proceed to the next highest correlation or return to the lowest state; the proper result is selected based on the current state of input triggers and output as the updated adaptive correlator state. When all three triggers are activated, a correlation lock is declared. The additional outputs *MaxIndex* and *ComplexCorrValue* pass along correlation results to the post-lock time and frequency offset estimation processes; in particular, *MaxIndex* provides the expected delay within the immediate correlation window (an integer 0 to 7) that may be refined by a weighted mass (center of gravity) calculator and ultimately added to *BaseIndex* to obtain the estimate $\hat{k_\star}$ for $k_\star$.

4.1.3.4.5 Adaptive Correlator Hardware Utilization

The adaptive correlator is one of the more resource intensive components in the chaotic receiver, but can be made more efficient by re-using the CMAC cells in demodulation mode for despreading the received signal. A summary of the adaptive correlator hardware utilization, segmented by functional subcomponents and including the dual-port acquisition buffers, is provided in Table 8.

TABLE 8. Hardware utilization for adaptive correlator.

Component	**XtremeDSP Slices**	**Registers**	**LUTs**	**BRAMs**
Acquisition Buffer	0	703	200	36
Center of Gravity	0	292	739	0
CMACs	24	5763	6122	0
Comparators	0	1527	473	0
Frequency Offset	3	567	1030	1
State Machine	0	85	844	0
Total	27	8937	9408	37

4.1.3.5 Adaptive Correlator Offset Estimation

At the conclusion of the adaptive correlator processing, time and frequency offset estimates are developed and fed forward for tracking loop initialization. To facilitate an accurate set of estimates, the starting index of the correlation relative delay k_0 is decremented by $\frac{N_{ma}}{2} - 1 = 3$ from the correlation lock $k_\star$ to re-center the suspected correlation peak into the center of the correlation window. Deriving an accurate time estimate is especially critical given the impulsive autocorrelation of the waveform. The time estimation method is based on a simple weighted beam center of mass calculation about the assumed correlation peak, while the frequency offset estimation tracks the suspected correlation peak during the final correlation to detect non-stationary deviations in correlation phase.

4.1.3.5.1 Time Offset Estimation

The center-of-gravity calculator treats the immediate correlation window of N_{ma} baseband sample delays (8 delays at 40 MHz sample rate, or 200 ns) as a beam with point masses $\{C_0, C_1, \ldots, C_{N_{ma}-1}\}$ at each of the N_{ma} relative delays. An offset in the beam indexing is induced and removed in order to retain weight C_0. The center-of-gravity is calculated as

$$\frac{(C_0, C_1, \ldots, C_{N_{ma}-1}) \cdot (1, 2, \ldots, N_{ma})}{\sum_{j=0}^{N_{ma}-1} C_j} - 1 = \frac{C_0 + 2C_1 + 3C_2 + 4C_3 + 5C_4 + 6C_5 + 7C_6 + 8C_7}{C_0 + C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + C_7} - 1$$

The resulting value is between 0 and 7 (non-integer), providing the peak location within the current N_{ma} baseband sample correlation window. To obtain the final time offset index, the output of the center-of-gravity calculator is added to the $\frac{N_{ma}}{2} - 1$ decremented *BaseIndex* to obtain $\hat{k_\star}$.

4.1.3.5.2 Frequency Offset Estimation

During the final N_3 correlation, intermediate correlation values are sampled and stored at 4 evenly spaced intervals to provide an estimation of the phase walk over time for the correlation peak. More precisely, the intermediate complex accumulation value is captured during the final correlation after 384, 768, 1152, and 1536 points and stored in a FIFO buffer as $PV[1:32]$, with later indices corresponding to earlier samples. The phase drift over time is then estimated from the expected correlation peak.

$$\begin{aligned}\phi_\Delta = \tan^{-1} \sum &(PV[2:5] - PV[10:13])^* \cdot (PV[10:13] - PV[18:21]) \\ &+ \sum (PV[10:13] - PV[18:21])^* \cdot (PV[18:21] - PV[26:29])\end{aligned}$$

Since the inverse tangent function is approximately linear for small arguments, and the phase drift of a locked correlation will necessarily be small, the inner product alone suffices as the angular drift estimate in a 384-baseband sample duration. Scaling this phase drift to a frequency offset estimate in the 9.6 μs time window, a multiplication by 16576,[43] the result is added to the triggered coarse frequency offset selected from $\{0, \pm 15, \pm 30\}$ kHz. The resulting frequency offset detection capability of the protoytpe chaotic communication system is approximately ± 45 kHz; simulation results have consistently demonstrated frequency offset estimates within 250 Hz of the actual frequency offset in moderate channel conditions, while experimental results in low SNR conditions yield accurate estimates within ± 500 Hz.

4.1.3.5.3 Offset Estimation Performance

[43] The actual scaling factor is $\frac{1}{2\pi\, 9.6\mu s} = 16578.64$, but may be efficiently implemented as a shift addition $2^{14} + 2^7 + 2^6 = 16576$.

During integration of the prototype chaotic receiver, it was determined that while the timing estimation mechanism works extremely well, the non-stationary chaotic spreading sequence limits the frequency estimator performance at low SNRs (approximately 6 dB worse than theory). In particular, the estimation window was narrowed such that the response begins to saturate at ±5 kHz instead of the desired $7.5kHz$. Estimator performance for both the timing offset estimator (simulated RMS timing error with 0 dB spread SNR, measured in chaotic chip durations) and frequency offset estimator (estimated versus actual frequency offsets) are shown in Figure 77. As expected, the timing offset estimator is optimal when the signal is centered in the N_{ma}-length correlation window. The non-idealities of the frequency estimator can be eliminated by performing a staged initalization of the tracking loops: timing loop, then frequency loop (lag term of second-order loop filter) and finally phase loop (lead term of second-order loop filter). Using that frequency estimation approach, the phase error detector output of the time synchronized waveforms is evaluated for phase drifts over multiple symbols (leading to the slightly longer preamble) to initialize the phase/frequency tracking.

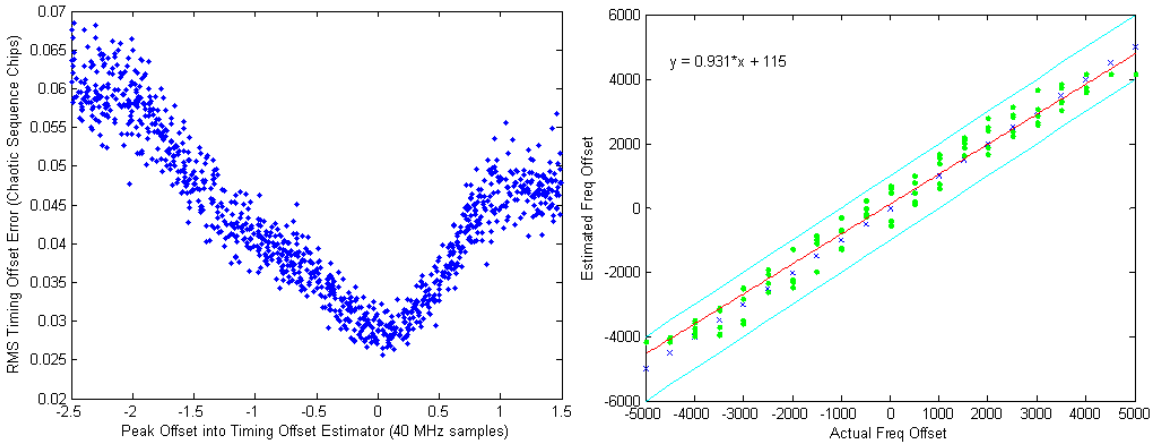

FIGURE 77. Adaptive correlator timing and frequency estimator performance.

4.1.4 Chaotic Waveform Synchronization

At the conclusion of acquisition, the adaptive correlator outputs a relative time delay that is parsed into an integer number of 10 MHz chaotic sequence chips, an integer number of 40 MHz baseband samples, and a fractional number of 40 MHz samples. These time controls are provided to the chaotic sequence generator, implemented as a sequence pause with the enable controls, a pair of selectable delay shift registers, and the Farrow resampler, respectively. Empirical measurements have shown the adaptive correlator time offset estimate to be accurate within 0.2 baseband samples, or 5 ns, at the end of these adjustments. A notional picture of the chaotic sequence time synchronization is provided in Figure 78, with blue lines indicating the various controls.

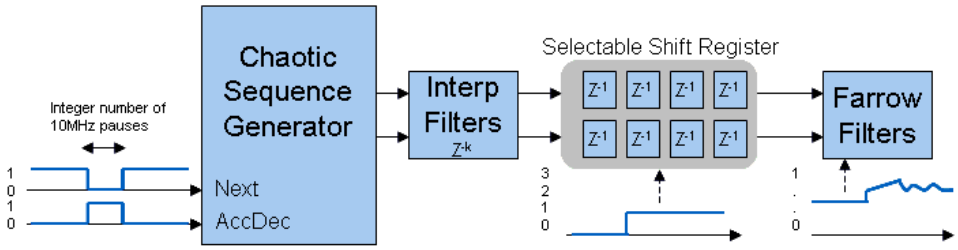

FIGURE 78. Parsing of chaotic sequence time synchronization.

Frequency synchronization is provided by jamming a scaled version of the frequency offset estimate into the lag term of the second-order loop filter. In addition, prior to beginning continuous phase tracking operations, a static phase offset is calculated on the time-synchronized waveform to jam in a lead term initialization. As an example, the soft symbol constellation (100 kHz samples) shown on the left side of Figure 79 captures the effects of frequency offsets and the static phase adjustment back into the first quadrant. The green dots correspond to the pre-acquisition and initial post-acquisition soft symbols, moving from an uncorrelated constellation (central region) to a time-synchronized, but phase-incoherent and drifting constellation (second quadrant, near $\frac{2\pi}{3}$), and finally to an approximately lock first-quadrant constellation point. The additional yellow, red, magenta, and black points will be described more thoroughly in terms of the phase loop performance, leading to the locked post-acquisition blue constellation points in the right figure.

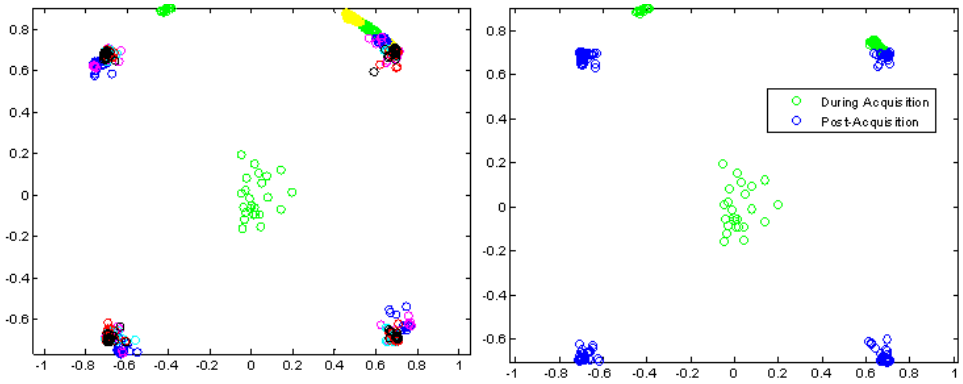

FIGURE 79. Phase loop initialization during worst-case frequency offset acquisition processing (left) and nominal acquisition processing (right).

4.2 Chaotic Receiver Signal Processing

The core signal processing for a coherent chaotic communication system, assuming access to a well synchronized chaotic circuit, follows the traditional models for direct sequence spread spectrum reception: the spread signal is received, converted to an oversampled baseband sample stream, despread to a collapsed signal spectrum, and then demodulated based

on the expected data constellation. Signal tracking is performed by a collection of early-late timing, phase rotation, and amplitude detectors that supply inputs to the time tracking loop, phase/frequency loop filter, and automatic gain control, respectively. The divergence from DS approaches begins with the despreading operation since chaotic chipping sequence samples represent complex conjugates with arbitrary phases as opposed to quadrature binary sequences; the despreading operation requires higher precision up to the ideal case of a hardware multiplier in lieu of a switched accumulation. Tracking of the despread signal is similar to DS approaches in the use of early-late detection and despread symbol phase error estimates, yet requires normalization of the nonstationary chaotic symbol energy to produce more stable results. Derivative observations of the symbol normalization technique allow implementation of a constant symbol energy technique to improve BER performance and a selective noise cancellation process that provides an effective increase in SNR. Most of these techniques were applied in the construction of the prototype receiver, although some represent improvements developed during the integration and test phases, verified through simulation.

4.2.1 Coherent Chaotic Waveform Demodulation

The central process in chaotic receiver demodulation processing is using the time synchronized digital chaotic circuit to despread the received signal, collapsing the spectrum down to the symbol bandwidth for insertion to a traditional demodulator. The discussion in this section will use examples focused on chaotically spread QPSK data, although the techniques apply to more general data constellations discussed in later chapters. A basic block diagram of a core chaotic receiver signal processing is shown in Figure 80.

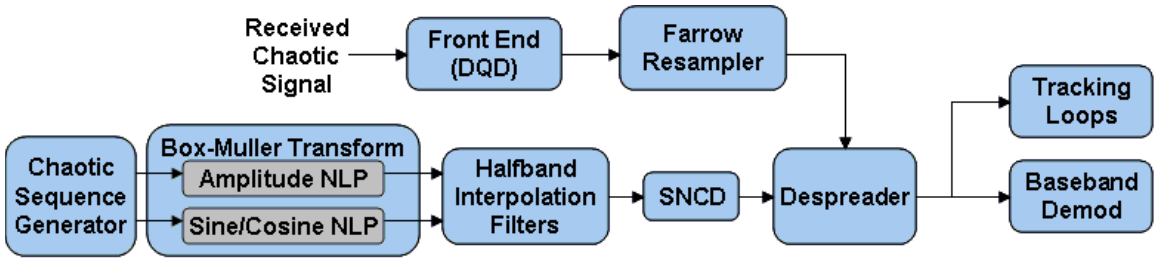

FIGURE 80. Block diagram of core chaotic receiver signal processing.

4.2.1.1 Coherent Chaotic Sequence Generation

As described previously, the fundamental barrier in constructing a coherent chaotic communication system is implementing a robustly synchronized chaotic circuit. The digital chaotic sequence is a deterministic process, reducing the synchronization of the chaotic circuits to a robust synchronization of the clock references, which may be largely solved via GPS references

or other stable references and a time tracking loop. The specific internal control mechanism of the chaotic sequence synchronization is proprietary[159, 160], but a brief qualitative description is included; all structures are identical to the chaotic sequence generator used in the transmitter.

As a summary, the sequence generator is globally enabled/disabled by an acquisition state machine, initialized to a predetermined chaotic base state, adjusted for the evolution of time between an agreed upon epoch and the current time, and then adjusted to bring the sequence output to a time-synchronized replica relative to the input of the despreader. First, a binary state load pulse replaces the current chaotic state with the RNS-defined calculated state vector. Then, a 20 MHz enable acts as a synchronous clocking mechanism to maintain a linkage between the chip clock (referenced through a DCM to a precision time reference) and the sequence evolution. Two chaotic sequence enable lines permit fine adjustments of the sequence to progress 0, 1, or 2 sequence values during the next clock cycle. The previous three signals are combined by making the 20 MHz enable an asynchronous control line operating at an integer multiple of the 20 MHz clock rate. The default configuration for the enable lines is chosen to indicate a linear progression of the sequence over time; the acquisition schema intentionally attempts to center the receive signal in the acquisition time window, requiring a controlled pause state on the internal chaotic sequence once the correct delay is calculated. The enable controls are also used to advance or retard the sequence as a result of time tracking. When larger jumps are required, a binary pulse strobes the ring generator, providing an input encoded into an RNS representation, and adds that to the current chaotic sequence state for the coarse time adjustment; after completing the state jump, the relative time uncertainty between the internal chaotic sequence and the received chaotic signal is less than half the acquisition time window, or 3.2 μs. Given the low levels of clock uncertainty that are available with GPS references, the clock drift during the initial transmission period ($\approx$ 100 μs) is dominated by propagation delay. A series of measured captures from the chaotic sequence synchronization using an RF loopback test configuration (no noise, permitting visual recognition of waveform characteristics), is shown in Figure 81.

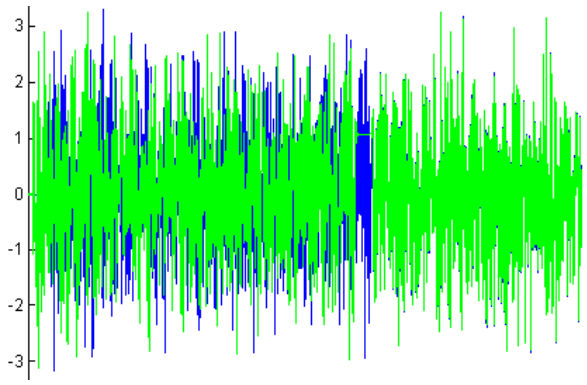

FIGURE 81. Synchronization of internally generated (green) and externally received (blue) chaotic signals.

A more detailed capture of the two sequences prior to synchronization, during synchronization (a pause in the internally generated sequence with evident halfband filter ringing), and after synchronization are shown in Figure 82. The two sequences are synchronized by applying the $(AccDec, Next)$ pulses and Farrow resampler controls.

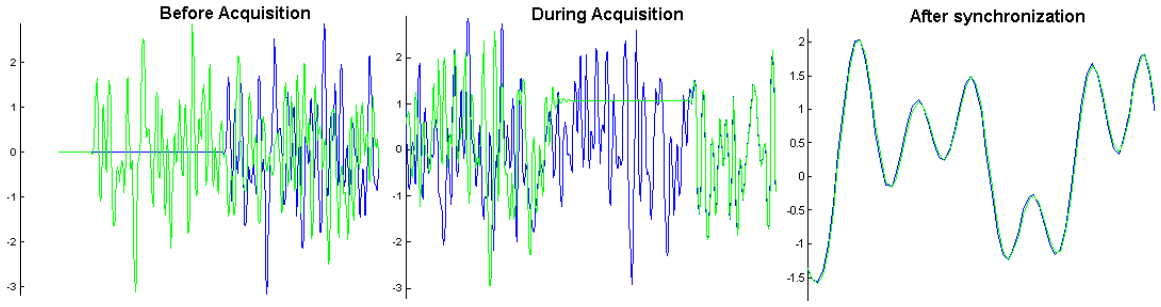

FIGURE 82. Detailed synchronization of internally generated (green) and externally received (blue) chaotic signals at various stages of synchronization processing.

4.2.1.2 Coherent Chaotic Despreader Design

Provided the time synchronized chaotic spreading sequence, the despreader combines the received chaotic signal with the conjugated internally generated chaotic sequence via a complex multiplication. Assuming infinite precision in this multiplication, the despreader output is then accumulated to obtain a symbol estimate as derived in the previous chapter.

$$E_{sym} = \sum_{k=0}^{T} x_k^* \cdot (y_k + n_k)$$

Practical hardware enforces tighter bounds on the numerical precision available; unlike direct sequence spread spectrum that uses binary values for x_k, the multiplication operation may not be accurately reduced to a selection between addition or subtraction of the received

signal samples.

$$E_{sym} = \sum_{k=0}^{T} x_k^*(y_k + n_k) \qquad x_k \neq \sum_{k=0}^{T} sign(x_k^*)\,(y_k + n_k) \qquad x_k \in \{-1, +1\}$$

It is expected however that lower bits in the binary representation contribute less to the separability or despreading of the signal from the background noise, leading to a potential trade between the computational complexity and despreader performance. The despreader implemented in the prototype chaotic communications receiver used 18-bit hardware multipliers, which preserve both the received and internally generated chaotic sequences completely. Lower precision implementation simplifies not only the despreader, but also the components conditioning the internal chaotic sequence and the received chaotic signal filtering; any reduction in despreader performance contributes to receiver implementation loss. Before evaluating the trade between performance and despreader precision, first note that any loss in precision will be preferred in the internally generated chaotic sequence since it is generated with approximately +60 dB of SNR, while the received chaotic signal is likely to be below the received noise floor at the input to the despreader. Therefore, consider a generalized separation of the despreader output with the internally generated chaotic sequence reduced to ones-complement binary format.

$$\begin{aligned} E_{sym} &= \sum_{k=0}^{T} x_k^* \cdot (y_k + n_k) \\ &\approx \sum_{k=0}^{T} -sign(x_k)\left(x_{k,2}2^2 + x_{k,1}2^1 + \ldots + x_{k,-14}2^{-14}\right)^* \cdot (y_k + n_k) \end{aligned}$$

Truncating the LSB from the representation of the internally generated chaotic sequence has minimal effect on the overall despreader output, justifying the use of ones complement binary representation. To evaluate this reduced precision despreading, let R be the number of MSBs retained in the representation of the internally generated chaotic sequence after coherent filtering, not counting the sign bit. As an example, $R = 3$ corresponds to the sign bit plus the three MSBs, such that all integer bits are either rounded (optimal) or truncated.

$$E_{sym,R=3} \approx \sum_{k=0}^{T} -sign(x_k)\left(x_{k,2}2^2 + x_{k,1}2^1 + x_{k,0}2^0\right)^* \cdot (y_k + n_k)$$

Four different cases of reduced precision despreading were considered:

1. Despreading with binary signed chaotic sequence
2. Truncation after R MSBs
3. Rounding to R MSBs

4. Scaled R MSBs

The first method represents despreading with the signed conjugate of the binary chaotic spreading sequence. The truncation method is also simple to implement in hardware and represents a straightforward selection of the sign bit plus the R MSBs. The third method is similar, using rounding to adjust the retained LSB. The fourth method uses a pre- and post-scaling to retain the (rounded) R MSBs starting with the first nonzero one. Two additional methods employing CSD-based pre-coding similar to a modified Booth algorithm[91] and a combination of CSD-coding/magnitude scaling were considered, but ultimately discarded since the combination of rounding and scaling gives such good results. Simulations of despreading operations on the chaotic received signal only,[44] measured as the reduction in despread symbol energy without noise, provide the receiver implementation losses shown in Figure 83. Including noise in the simulations should have a similar effect in all cases since the truncation and rounding operations do not appreciably change the internally generated signal variance.

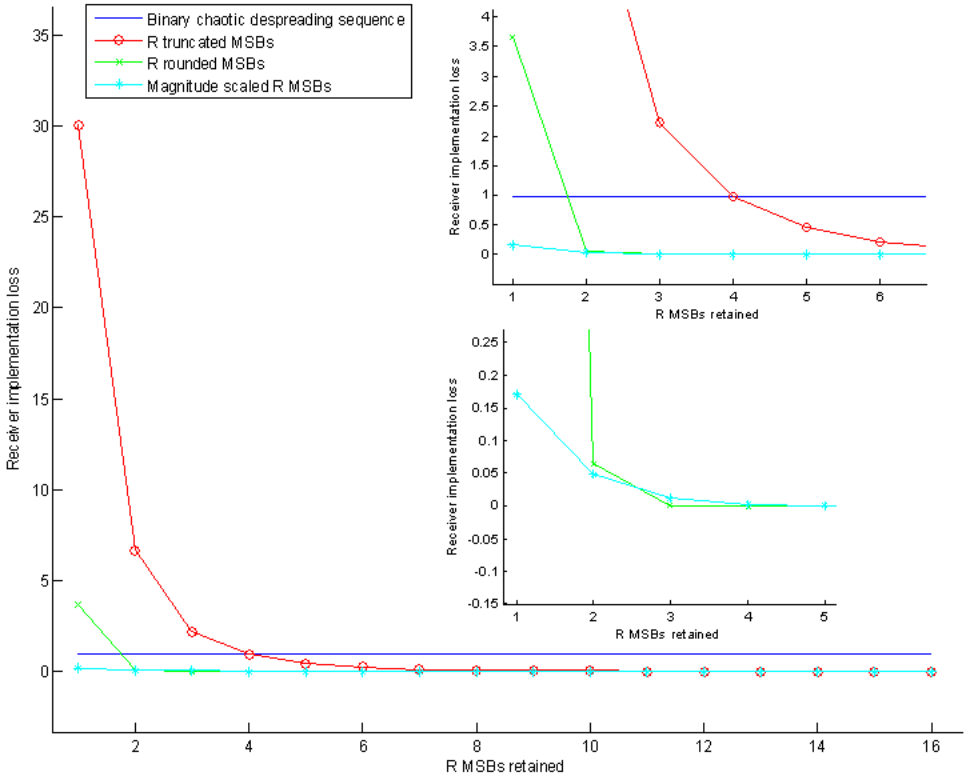

FIGURE 83. Simulated performance of reduced precision chaotic signal despreading.

The simplest option of using a binary despreading sequence is feasible with approximately 1 dB of associated implementation loss; recognition that this level of imprecision is allowable will make the internal signal processing much more efficient by trading components downstream

[44]Matlab scripts of reduced precision despreading characteristics are included in the appendix.

or implementing the despreader with many fewer resources. The best option of the four is to use either the rounded or rounded and scaled values with either 2 or 4 MSBs implemented as a static shift addition in lieu of hardware multipliers; both options result in less than 0.1 dB receiver implementation loss, while the 4 MSB case will extend better to extremely low received signal SNR scenarios and provide additional flexibility in preceding signal processing.

4.2.1.2.1 Box-Muller Approximations

By virtue of coherency, the Box Muller transformation should be configured to generate an identical sequence to that used at the transmitter. The chaotic sequence must be produced with identical operations to support the constant-energy symbol transmission and various multiple access techniques discussed later that accumulate errors over a long period to change symbol clocks or other parameters. The nonlinear processor implementation is already sufficiently efficent, with the potential exception of the two multiplications combining the Rayleigh magnitude NLP and quadrature phase NLPs; exchanging those multipliers with a low-order shift-addition structure is possible, yet requires an additonal approximation for the quadrature magnitude calculation for threshold comparisons.[45]

4.2.1.2.2 Coherent Chaotic Receiver Filtering

The ideal filtering approach at the chaotic receiver is a mirror image of that used at the transmitter, ensuring that the two time-synchronized signals are as identical as possible. Noting that the despreader operates acceptably when errors are induced in the LSBs of a reduced precision multiplication, the receiver filtering constraints are considerably lessened. In particular, the choice of filtering may be simplified such that any unwanted spectral content is below that of the received noise, accounting for despreading effects; in most cases, this requires at most 4-bits of accuracy ($\approx$ 24 dB). Consider again the filtering shown in Figure 80: the halfband interpolation filters are required to increase the sample rate of the chaotic sequence from the generated 10 MHz to the received signal baseband sample rate. The prototype chaotic receiver uses a baseband sample rate of 40 MHz, leading to a 4x interpolation that was broken into two 2-phase halfband interpolators with numerical precisions similar to those used in the transmitter. These precisions may be reduced significantly provided a 30-40 dB rejection is maintained. Re-designed filters to replace those used in the prototype receiver were developed

[45] One example is the approximation $|x| \approx \max \mathbb{R}(x), \mathbb{I}(x) + \frac{3}{8} \min \mathbb{R}(x), \mathbb{I}(x)$. Any approximation that is used must be completely coherent at the transmitter and receiver.

at a hardware savings of approximately 80%. Similar improvements were achieved with reductions of the continuous phase Farrow filter.

4.2.1.2.3 Selective Noise Cancellation

A significant benefit of the well synchronized coherent chaotic sequence spread waveform is the ability to predict the instantaneous sample energy a priori at the receiver on a sample-by-sample basis. Since the soft symbol estimations are created from a despreading process that consists of conjugate multiplication and accumulation, knowledge of instantaneous sample energy may be used to selectively discard the samples having a relatively low amount of instantaneous energy[158]. Moreover, discarding a sample of the received signal also results in discarding the corresponding background noise contribution to the soft symbol estimate from that sample; both signal and noise power are being reduced by discarding the sample. Under assumptions that the received chaotic waveform and background are independent samples of additive Gaussian white noise and that the expected noise power is greater than the expected signal power ($|\alpha| < 1$), the expected noise energy discarded in a sample is a constant, while the expected signal energy is related to a deterministic process and may be selectively chosen to be much less than the instantaneous noise energy.

Consider the received chaotic sequence as having a Normal statistical distribution with characteristics

$$N(200|\alpha|, \sqrt{200 \cdot (1 + 2|\alpha|^2)}^2)$$

If, by having knowledge of the instantaneous sample energy, all samples with magnitude less than $0.25|\alpha|$ are discarded, the modified statistical distribution will be approximately Normal with a large impulse of probability density at 0 as shown (normalized to be independent of $|\alpha|$) in Figure 84. The magnitude of the impulse is equal to the integrated probability density over (-0.25,0.25).

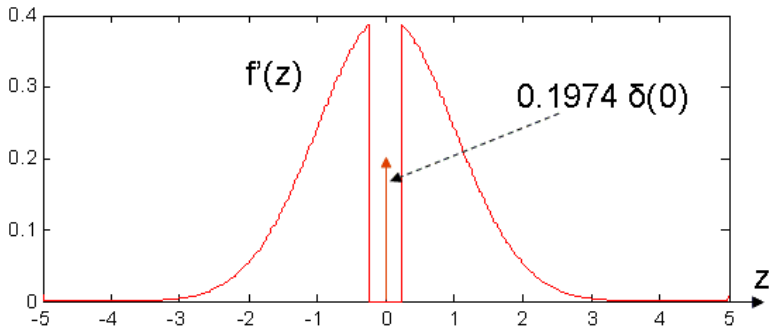

FIGURE 84. Normal distribution with collapsed central region.

Performing the despreading process using a "discarded" sample of value zero results in zero addition to the signal and noise energies. The statistical expectation of the independent noise power within any sample is a constant, while the discarded signal is known to be of magnitude less than $\frac{|\alpha|^2}{16}$. The effective normalized statistical distribution is shown in Figure 85.

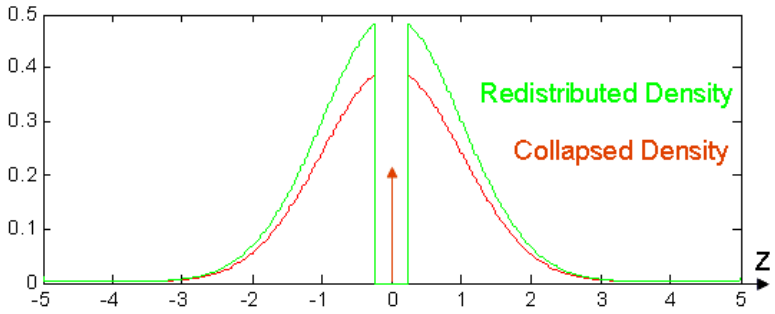

FIGURE 85. Re-distributed standard normal distribution.

The effective signal-to-noise ratio can be obtained by comparing the effects of the despreading process after samples are selectively discarded. The expected signal energy is obtained from an expectation of the squared redistributed probability density, while the noise energy is reduced approximately linearly due to the extraction of the central mass in the signal distribution. As an example, consider a normalized BPSK chaotic waveform that has spread power 10 dB below the noise power ($|\alpha| = \frac{1}{\sqrt{10}}$); using the distribution above for selective noise cancellation, discard all samples with signal voltage of magnitude less than $\beta = 1.0$. The noise energy is not reduced by 68%, since it must be evaluated versus the conditional signal distribution (approximately uniform on [-1,1], resulting in an approximate expected magnitude of $\frac{1}{2}$); the overall noise reduction is therefore around 30%. The reduction in signal power, however, is equal to the energy over [-1,1] within the standard normal distribution, which is closer to 20%. Therefore, the SNR of the soft symbol estimate improves by the ratio 0.8 / 0.7 = 1.14 or 14%, which is an effective increase of 0.6 dB in the carrier level. Note that this punctured distribution is strictly controlled by the selection of β and does not depend on input signal level when $|\alpha|$ is sufficiently small.

More precisely, let z be the argument of a standard normal distribution, $f(z)$ be the probability density function (*pdf*), and $F(z)$ be the cumulative density function (*cdf*). The selective noise cancellation technique replaces $f(z)$ by a statistical distribution of the form

$$\hat{f}(z) = \begin{cases} (F(\beta) - F(-\beta))\delta(z) & |z| \leq \beta \\ f(z) & |z| > \beta \end{cases}$$

This modified distribution approximates the standard rules for a combined discrete and continuous probability distribution, namely

$$\int_{-\infty}^{\infty} \hat{f}(z)\, dz = 1 \qquad\qquad \hat{f}(z) \geq 0 \,\forall z$$

The effective reduction in signal energy and noise energy caused by puncturing the internally generated chaotic sequence may be calculated using a Chi-square distribution with one degree of freedom for a BPSK modulated chaotic waveform and a Chi-square distribution with two degrees of freedom for a QPSK modulated chaotic waveform.

$$\text{Residual signal energy: } = \int_{-\beta}^{\beta} \frac{z^2}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}\, dz = erfc\left(\frac{\beta}{\sqrt{2}}\right) + \sqrt{\frac{2}{\pi}}\beta e^{-\frac{\beta^2}{2}}$$

simplifying the closed form solution[161]

$$\int u^2 e^{-\frac{u^2}{2}} du = \sqrt{\frac{\pi}{2}} erf\left(\frac{u}{\sqrt{2}}\right) - u e^{-\frac{u^2}{2}}$$

The corresponding reduction in noise energy in a soft symbol comes from the selective cancellation of internally generated chaotic sequence samples that puncture the received signal (containing both signal energy and noise energy) to remove an approximately linear amount of energy. More precisely, the expected residual noise variance is approximately

$$\begin{aligned}
\text{Noise Variance: } &= \sqrt{E[x^2 n^2 \,|\, |x| \geq \beta]} \\
&= \sqrt{E[x^2 \,|\, x \geq \beta] \cdot P(|x| \geq \beta) \cdot \sigma_n^2} \\
&= \sqrt{\left(erfc\left(\frac{\beta}{\sqrt{2}}\right) + \sqrt{\frac{2}{\pi}}\beta e^{-\frac{\beta^2}{2}}\right) \cdot erfc(\beta) \cdot \sigma_n^2}
\end{aligned}$$

To validate this computation, consider again the case of $\beta = 0.25$, which produced an estimated reduction of 30% in the noise energy, compared to the calculated 27.6% reduction. Defining more succinctly the residual energy in the signal $g_1(\beta)$ and noise $g_2(\beta)$,

$$\begin{aligned}
g_1(\beta) &= erfc\left(\frac{\beta}{\sqrt{2}}\right) + \sqrt{\frac{2}{\pi}}\beta e^{-\frac{\beta^2}{2}} \\
g_2(\beta) &= \sqrt{\left(erfc\left(\frac{\beta}{\sqrt{2}}\right) + \sqrt{\frac{2}{\pi}}\beta e^{-\frac{\beta^2}{2}}\right) \cdot erfc(\beta)}
\end{aligned}$$

a scaling factor $G(\beta)$ may be created from the ratio of the residual symbol energy and residual noise energy.

$$G(\beta) = \frac{g_1(\beta)}{g_2(\beta)} = \sqrt{\frac{erfc\left(\frac{\beta}{\sqrt{2}}\right) + \sqrt{\frac{2}{\pi}}\beta e^{-\frac{\beta^2}{2}}}{erfc(\beta)}}$$

This expression for $G(\beta)$ is divergent for large β, which is consistent with both a large signal-to-noise ratio and a very small number of samples; emperical observations has shown

that $G(\beta)$ is accurate up to values in the neighborhood of $\beta \approx 1$, after which tracking loop errors, imperfect timing synchronization, and environmental interferers dominate any potential processing gains. Observing $G(\beta)$ in its limiting conditions when practical operating ranges are incorporated,

$$\lim_{\beta \to 0} G(\beta) = 0 \qquad \lim_{\beta \to 0} \frac{\partial}{\partial \beta} G(\beta) > 0 \qquad \lim_{\beta \to \infty} G(\beta) = -\infty \qquad \lim_{\beta \to \infty} \frac{\partial}{\partial \beta} G(\beta) < 0$$

to indicate that initially the performance increases up to the point that the number of samples in the symbols reduces below a meaningful point, leading to reductions in performance; by use of the intermediate value theorem on the derivative of $G(\beta)$, there is expected to be a local maximum somewhere in the interior of β.[46] Emperically, the optimal value of β has been determined to be in the range $\hat{\beta} \in (0.5, 1.5)$, although different assumptions on timing uncertainty and phase jitter, loop dynamics, and spreading ratio will likely cause $\hat{\beta}$ to change. At a value of $\beta \equiv 1$, the simulated processing gains was approximately 3.0 dB versus the calculated $G(\beta = 1) = 3.53$ dB.

4.2.1.2.4 Timing and Phase Jitter Susceptibility

The despreader performance analysis to this point has assumed a sufficiently time- and phase-synchronized replica of the received chaotic waveform being produced by the receiver digital chaotic sequence generator. These assumptions have been proven reasonable both analytically and emperically, yet the performance of the despreader is degraded when either deviates from perfect synchronization. Timing jitter manifests in lower despreader correlation values since the actual delay differs from the ideal delay; the impulsive correlation of the waveform exacerbates large jitters, resulting in increasing implementation losses. Phase jitter becomes more relevant in discussion of the phase error detector and phase/frequency loop filter, while a low susceptibility to phase jitter reduces the requirements on the sine/cosine NLP used in the Box Muller transformation. To bound the performance degradations caused by timing and phase jitter, an analysis and simulation were performed assuming Gaussian distributed errors of either timing or phase jitters; note that the chaotic signal is considered unlocked whenever the perturbation from ideal time delay exceeds one chipping sequence duration as a result of the impulsive autocorrelation (assuming extreme 4σ events occur approximately once every 10 seconds, the receiver loses lock for RMS timing jitters exceeding $\frac{1.5}{4} = 0.375$ chaotic

[46] Existence of a local maximum does not guarantee that there are not multiple local maxima/minima, yet the expected monotonicity of the $G(\beta)$ and performance degradation curves leads to a suspected single maximum.

chips). The results of these analyses are shown in Figure 86.[47] Practical phase jitter values are on the order of 2-3 degrees, while emperical evidence of the prototype chaotic timing loops shows less than 0.05 baseband samples of timing drift jitter.

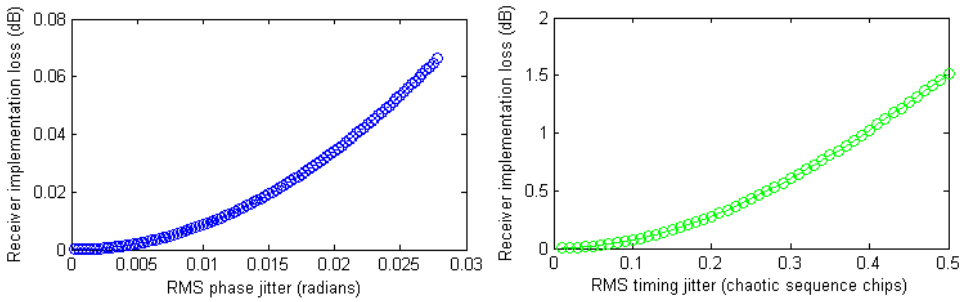

FIGURE 86. Phase (left) and timing (right) jitter despreader susceptibility.

As depicted in these plots, the receiver implementation loss caused by timing and phase jitter are expected to be relatively small when robust time synchronization is maintained.

4.2.1.2.5 Early-Late Detection Mechanisms

The final topic covered with regards to despreader design is the early-late detection format used for calculation of phase and timing errors. Phase error estimators are relatively simple to design and require a priori knowledge of the data constellation such that the error estimates are derived from soft symbol estimates. Any closed loop negative feedback mechanism that prevents runaway frequency errors (i.e. unconditionally stable loops) will be sufficient for the chaotic receiver; simulation and emperical measurements show that error estimate accuracies on the order of 1 degree will support a robust design. Timing errors require more in-depth control due to the impulsive autocorrelation of the chaotic waveform; choosing the early-late detection mechanism too narrow will result in a loss of ability to perceive the timing drift (reducing the effectiveness of the timing error detector), while choosing the early-late detection mechanism too wide leads to potentially slow responses and narrow tracking capability. Traditional direct sequence time tracking loops use $\pm\frac{1}{2}$ chipping sequence delays as a baseline early-late detector[162], while improved approaches including binary offset code (BOC) modulated spread carriers permit more comprehensive methods[163, 164]. Other early-late detectors create more robust estimator by the use of additional early-early and late-late detectors or variable width early-late detectors. One of the basic desired properties is that the error estimator be linear about the central correlation peak. Another characteristic is that

[47]The simulations are performed as long-term evaluations at the despreader sample rate although timing and phase jitters operate at the loop update rate; the values of the results will be identical even though the variances will not be.

the detector recognize when the peak is slipping in one direction, with loop gain sufficient to prevent the correlation slipping out of lock. Finally, the detector will be easier to implement when the early-late steps are an integer number of samples, being $\{1, 2, 3, 4\}$ with a 10 MHz spreading sequence rate and a 40 MHz baseband sampling rate. To demonstrate the range of values and marginal effects of changing the spacing about the correlation peak center, a plot of symmetric detector values δ over the range of ± 1 chaotic chip timing deviation are shown in Figure 87; note that timing lock is lost as soon as the slope of the detector changes sign, leading to positive feedback.

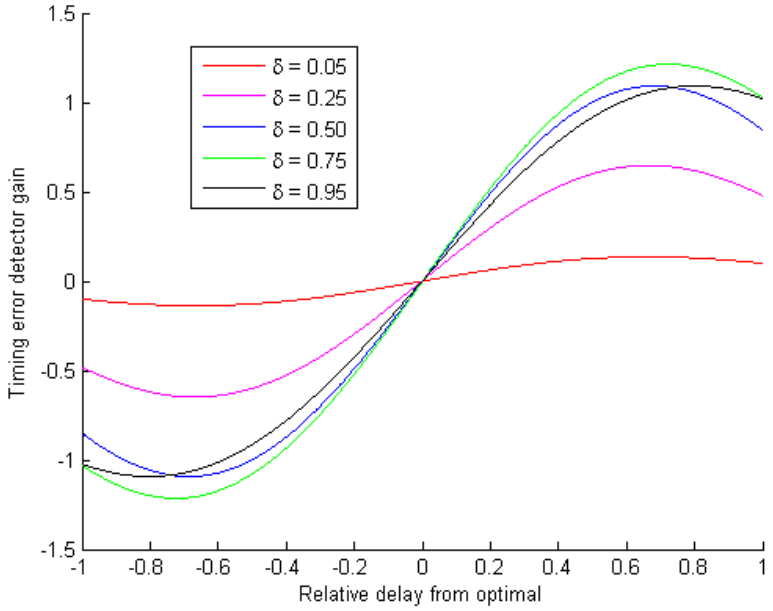

FIGURE 87. Timing error detector response for a range of symmetric early-late delays.

The prototype chaotic despreader was developed using the traditional $\pm\frac{1}{2}$ chip delays, corresponding to ± 2 40 MHz baseband sample delays. Arguably, the optimal detector uses the δ corresponding to the maximal detector slope over $\pm\frac{1}{2}$ chip durations, which was determined via curvefit to be ± 0.675 chaotic chip durations.

4.2.1.2.6 Chaotic Despreader Summary

The overall output of the chaotic receiver despreading mechanism is a stream of chip values that coherently combine during the symbol interval. For a chaotically spread QPSK data constellation, each of the I/Q channels are expected to have roughly identical streams of Rayleigh distributed random variable that accumulate to an approximately constant soft symbol. The measured despreader output before accumulation (real and imaginary) at 0 dB spread SNR is shown in Figure 88.

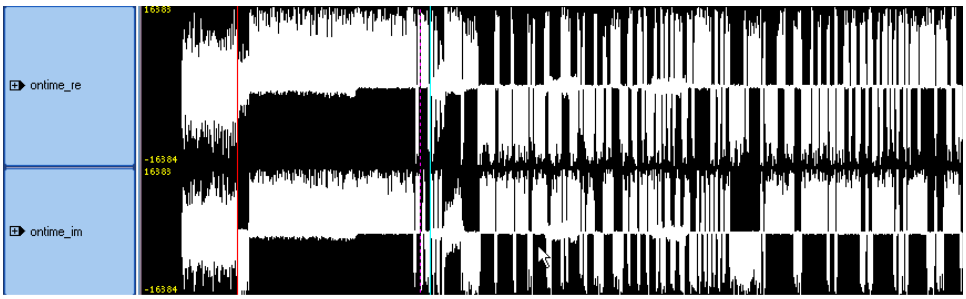

FIGURE 88. Chaotic receiver despreader output before accumulation.

The despreader starts out with incoherent noise (distributed according to a scaled Bessel function) before acquisition lock (timing lock and coarse frequency lock) is achieved. This noise is followed by a brief period where the receiver is allowed to zero in its frequency lock and then a static phase correction is implemented in the NCO to force the output into the first quadrant. Phase tracking is then initiated, with the remainder of the acquisition preamble used to further discipline the tracking loops. At the conclusion of the preamble, a pair of diasmbiguity bits are used to decipher spectral inversions, followed by received data symbols. The data symbols are evident by the short bursts of despreader output that coherently add to a soft symbol estimate. A clearer depiction of this is possible with the simulated despreader output (0 dB spread SNR) shown in Figure 89, with I/Q channels as blue/green outputs and the symbol clock as a one-sample red pulse.

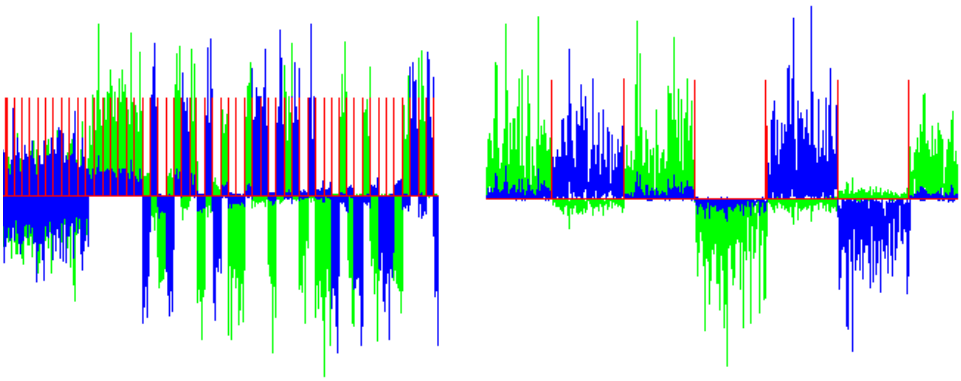

FIGURE 89. Chaotic receiver despreader output before accumulation.

A summary depiction of the post-accumulation despreader output is shown in Figure 90, starting with a long run of the accumulated despreader output, a close-in view of the despreader output during the acquisition preamble, a histogram of the soft symbol decisions, and an arbitrary selection of the accumulated soft symbols.

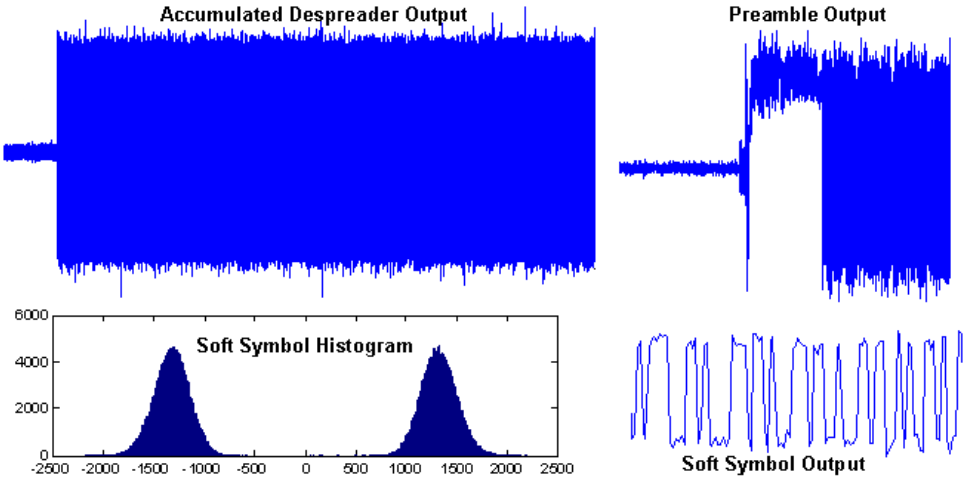

FIGURE 90. Chaotic receiver despreader output after accumulation to soft symbols.

4.2.1.3 Chaotic Symbol Normalization

One of the practical downsides of receiver processing for a chaotic waveform that uses a chaotic spreading sequence is a significant non-stationary symbol energy that occurs independent of any automated gain control. Quite simply, the symbol energy changes on a symbol-by-symbol basis due to the emulated random generation process. The statistical distribution for a soft symbol estimate during demodulation mode is approximately a Gamma magnitude at an arbitrarily chosen phase representing data content. The amplitude distribution of a sequence of symbols, demodulated from 0dB spread SNR are shown in Figure 91, relative to an approximate Gamma distribution.

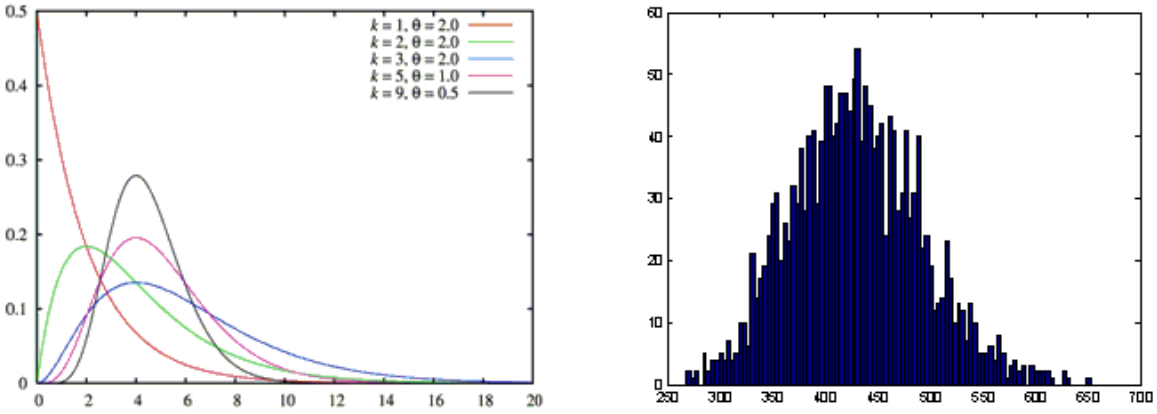

FIGURE 91. Comparison of ideal Gamma distribution (left) to received soft symbol distribution (right).

For PSK baseband data modulation types, this non-stationary amplitude characteristic is largely negligible: hard decisions are made by sign alone, in which case the background noise effects have much more effect than the scaling. For amplitude modulated baseband data types

(e.g. 16QAM, 16APSK, etc), this amplitude scaling is significant since the natural amplitude variation compounds the error rate probabilities by potentially combining with the random noise contribution. A notional signalling constellation of the pre- and post-normalization soft symbols for QPSK and 16QAM is shown in Figure 92.

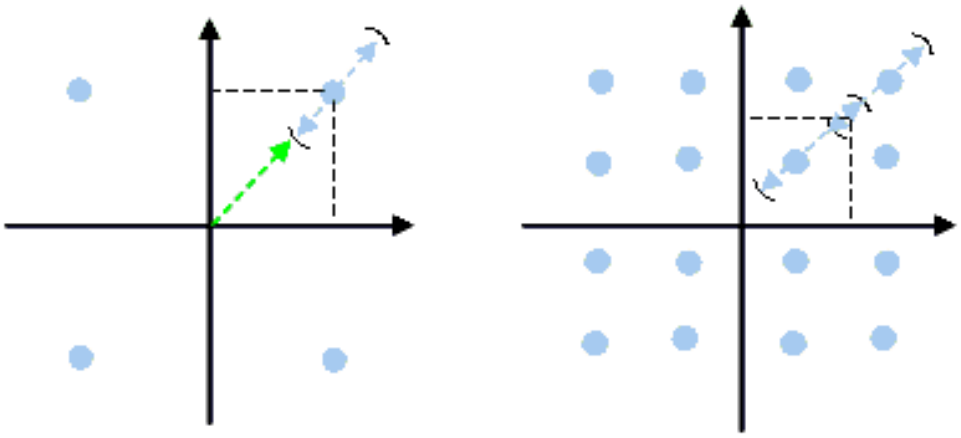

FIGURE 92. Effects of non-stationary chaotic soft symbol estimates for QPSK (left) and 16QAM (right).

To convert this non-stationary soft-symbol amplitude into an approximately stationary one, a symbol-by-symbol adjustment[165] is made by normalizing the soft symbol estimate by the predicted normalized energy in that symbol. Rather than a conventional statistical expectation which returns a mean of the Gamma distribution, this "predicted normalized energy" refers to the a prediction using the internally generated chaotic spreading sequence that is time and phase locked with the carrier tone of the transmitter. Data modulation that rides on top of this carrier (both non-stationary symbol amplitudes and phase rotations) is only known at the transmitter, so cannot be used in the prediction. Instead, the symbol energy, assuming a practically stationary carrier amplitude (as tracked using a traditional lowpass filtered AGC loop), is estimated from the symbol power of the internally generated sequence despread with itself. A representative QPSK constellation output of the prototype chaotic communications simulation both before and after symbol-by-symbol normalization is shown in Figure 93.

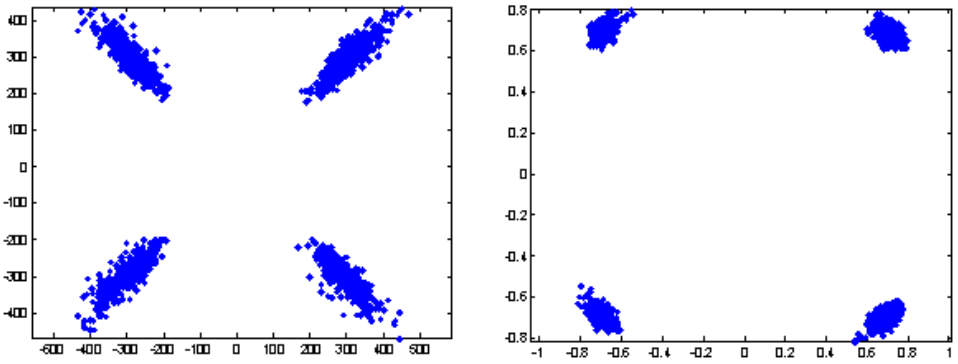

FIGURE 93. Comparison of chaotic symbol estimates before (left) and after soft symbol normalization (right).

This technique will be most beneficial when considering lower spreading ratios or higher capacity data constellations, such as would be most useful in multiple access chaotic communications. Using a similar technique is absolutely necessary for AM-based constellations. As an example, consider the comparative despread constellations of a chaotically spread 16QAM data constellation at -7 dB despread SNR (red), -2 dB despread SNR (green), +3 dB despread SNR (blue), +8 dB despread SNR (black) and +33 dB despread SNR (cyan) in Figure 94.

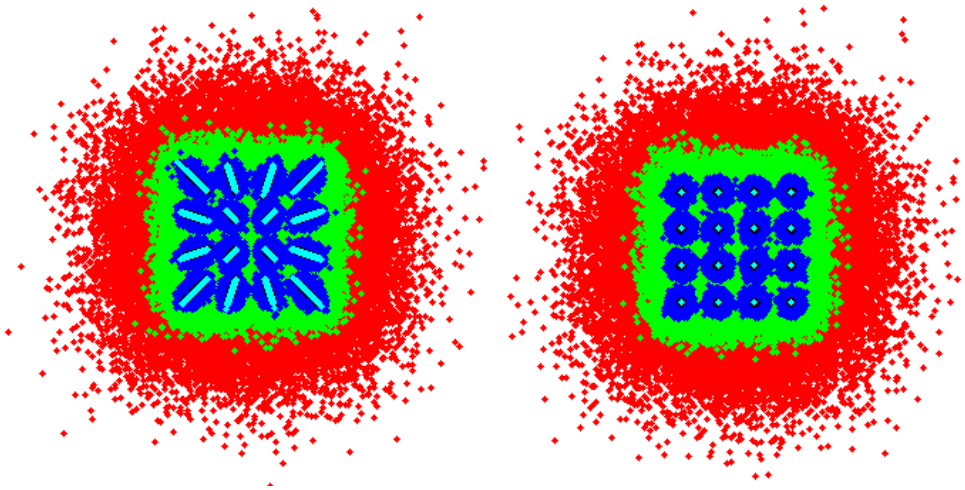

FIGURE 94. Despreading of chaotically modulated 16QAM with (right) and without (left) symbol normalization at various SNR levels.

4.2.1.6 Constant Energy Chaotic Symbol Modulation

A second method exists for improving the BER performance by varying the duration of the transmitted pulse (ignoring any amplitude levels) such that the expected normalized symbol energy transmitted is a constant. Transmitting a constant energy each symbol, as is inherent to conventional communication systems, leads to optimal BER performance. To implement this duration dithering approach, an accumulator is attached to the output of the

chaotic sequence generator to compute in real time the expected symbol energy; once this value exceeds a pre-defined threshold, the asynchronous symbol clock enable is pulsed, moving the transmitter to the next data symbol. A depiction of the hardware implementation is shown in Figure 95.

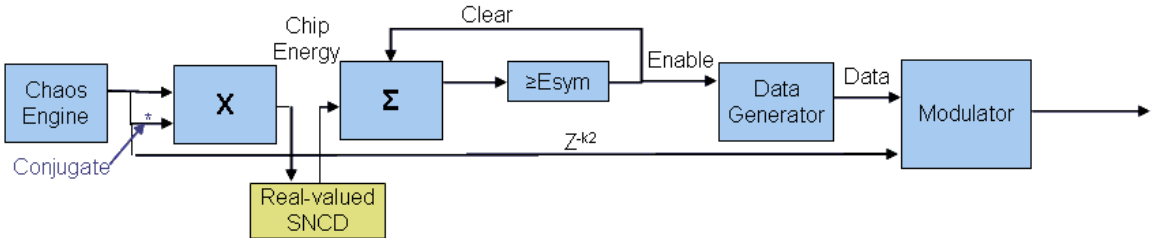

FIGURE 95. Transmitter modification for constant energy chaotic modulation.

As an example, consider a threshold chosen to represent the average energy in 200 quadrature AWGN samples before and after energy normalization at the transmitter. The expected energy per symbol is thus expected to be normalized to a much tighter distribution; for large spreading ratios, the non-stationary effects of the varying amplitude spreading chips will tend to average out, making the technique most beneficial for lower spreading ratios or higher data capacity modulations; this will be a significant improvement for multiple access chaotic communications implementation. To demonstrate this effect, a random selection of 8 million chaotic sequence values was used to modulate data based on constant symbol durations (green) and constant symbol energy (cyan) as compared to the ideal constant symbol energy (blue) in Figure 96. The diagrams to the right show an approximate Gaussian distribution for each of the free parameters, either the symbol energy contained in a constant duration symbol of the duration of the constant energy symbol.

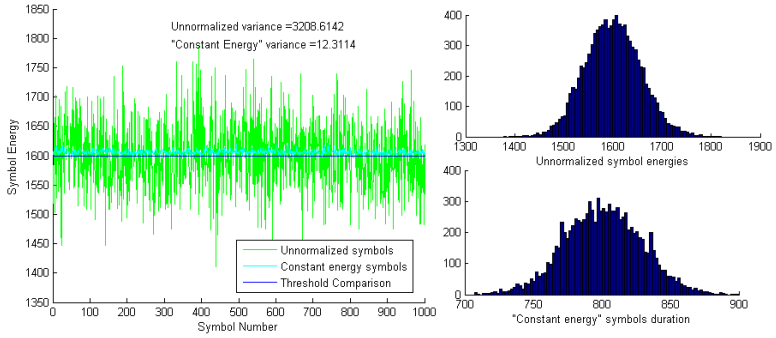

FIGURE 96. Comparison of constant duration (green) and constant energy (cyan) chaotic modulation.

A more general approach to the second method is an adaptive data rate algorithm that varies the symbol energy per symbol in response to any protocol feedback mechanism. The

transmitter can optimize the data rate based on specific BER levels by varying Esym; this value of Esym could easily be commanded by a communications controller and adapted over time based on transmission environment. Additional applications of this technique are included in Chapter 9.

4.2.2 Chaotic Waveform Tracking

Once the chaotic signal is acquired and the tracking loops are initialized, the chaotic receiver transitions into a steady-state tracking and demodulation mode. The initial tracking is performed on the remainder of the acquisition preamble with a wider tracking bandwidth when the received signal is a chaotically modulated CW tone, transitioning to narrower tracking after completion of the preamble. Hardware utilization between the various modes is reduced via selected re-use of the CMAC cells inside the adaptive correlator for the despreader and loops. This section analyzes the time, phase/frequency, and gain control tracking for the prototype chaotic receiver, extending the results to general guidelines for chaotic communication systems.

4.2.2.1 Chaotic Waveform Time Tracking

Robust steady-state time tracking of the chaotic waveform is certainly the most important characteristic of the prototype receiver, since the impulsive autocorrelation of the waveform ensures loss of lock after small drifts. One desire for the time tracking is to build in a timing drift momentum factor, leading to higher order tracking loops, in order to maintain lock during brief fades caused by multipath or interference. During those outages, the lead term contribution will ideally average out to zero. Further, the timing loops must prevent a small number of faulty time error estimates from desynchronizing the signals. Finally, the time tracking mechanism must incorporate the despreader early-late gate design discussed previously into loop gain calculations.

The prototype chaotic receiver uses a loop update rate (LUR) of 100 kHz and must be able to compensate for the cumulative range of timing drifts, primarily from unlocked oscillators and physical motion between the transmitter and receiver that changes the propagation delay. Oscillators used for clocking the digital processing at the transmitter and receiver are ideally locked via an absolute reference or a DCM, yielding drifts of a few cycles; for decent oscillators this drift is bounded by 1 ppm or 10 ppm, corresponding to 0.0001 - 0.001 chaotic chips per loop update. Approximate upper bounds for motion are 160 kph for most ground vehicles and 1000 kph for most airborne vehicles, corresponding to a head-on drift rate of 148 ns/s and 926 ns/s, respectively. During a single LUR duration, those drifts translate to $1.5 \cdot 10^{-5}$ and

$9.3 \cdot 10^{-5}$ chaotic chips, which is well below that of the potential oscillator drifts.

4.2.2.2 Chaotic Waveform Phase/Frequency Tracking

Phase and frequency tracking of the chaotic signal are more forgiving than time tracking. Three different phase error mechanisms were considered for the prototype chaotic receiver, with the chosen one being a straightforward phase error estimate of the despreader output (compared to the expected signal constellation) fed into a second order loop filter. The second method, a nonlinear 5^{th}-order construct, showed better performance in simulation, but cannot be shown to be unconditionally stable. The third method extends the basic phase error estimator to use of the early-late despreader outputs in addition to the prompt output. A loop bandwidth of 1 kHz was chosen for steady-state phase and frequency tracking, although a wider tracking bandwidth is enabled during the preamble and any periodic ambles. Loop integration yielded the expected performance when the loop bandwidth was widened to approximately 1.2 kHz, with the difference believed to be a result of the residual variance of I and Q despreader outputs.

4.2.2.3 Chaotic Waveform Gain Control

Gain control in any communication system supports maintaining an acceptable SNR for signal reception. In spread spectrum communication systems, the automatic gain control (AGC) function focuses on the receive energy that consists mostly of the background noise. Traditional direct sequence receivers often attempt to maintain a $\frac{1}{4}$-full scale input to the A/D converter[154] to prevent saturation, while simulations of the spread chaotic waveform show that a slightly greater backoff may provide better performance ($\frac{1}{5}-$ to $\frac{1}{6}-$full scale. The AGC loop update was chosen significantly slower than the 100 kHz LUR, allowing it to provide slow corrections to the received signal level. The block downconverter[144] used in the prototype chaotic receiver accepts a gain level in 1 dB steps over a 50 dB range to support this AGC function.

4.3 Chaotic Communications Summary

The present chapter has presented the detailed analysis and exemplary methods for constructing a practical coherent chaotic communication system using a discrete-time discrete-amplitude digital chaotic circuit. The robust chaotic circuit synchronization of distinct replicas demonstrates functionally approaching Percora's identical synchronization, ultimately resolving the last major barrier to practical and efficient chaotic communication systems. These

digital chaotic circuits in their simplest form are chaotic-based PRNGs implemented with stable clocks; the actual spreading sequence may be implemented without "chaotic" properties and still retain maximal entropy transmission provided the PRNG has a long enough cycle length. The RNS-based chaotic sequence generation structure however provides an efficient mechanism for manipulating the spreading sequence state. Combining this synchronized digital chaotic circuit with extensions of direct sequence spread spectrum communication systems, a practical approach to chaotic communications was demonstrated. Various novel techniques inherent to the chaotic waveform, including selective noise cancellation and spread data symbol energy control were developed and combined with more general approaches to adaptive correlation to make the chaotic communication system nearly as efficient as traditional DS approaches.

In summary, the current and preceding chapter fundamentally demonstrate the successful analysis, design, and implementation of what is believed to be the world's first practical coherent chaotic communication system. The methods described in these chapters may be extended from this fundamental chaotic waveform to what appears to be a new class of maximal entropy waveforms having broad applicability. Subsequent chapters address the broader categories of distortion mitigation in chaotic communication systems, PAPR-adjusted maximal entropy waveforms, generalized chaotic modulation of arbitrary digital data constellations, and chaotic multiple access communication systems.

Chapter 5: Distortion Mitigation in Chaotic Communications

One of the fundamental benefits of a spread spectrum waveform is the intrinsic multipath mitigation that provides signal separation capabilities at time intervals near the spreading sequence rate rather than the baseband data rate. Chaotically spread waveforms have an additional advantage over traditional DS spread waveforms in that the autocorrelation function is almost perfectly impulsive without sidelobes common to many direct sequence waveforms. In addition, direct sequence waveforms contain embedded cyclostationary effects caused by the periodicity of the DS spreading sequence that ultimately reduce signal entropy and are visible as increased correlations at integer numbers of data symbol periods. Improvements in DS multipath performance can be obtained by increasing the code repetition period or spread bandwidth; a simple example is the difference in GPS ranging capability between C/A and P codes. This chapter presents a brief comparison of the multipath characteristics for direct sequence and chaotic sequence spread communication systems, followed by a qualitative analysis of chaotic communications performance through fading channels and in the presence of interferers. The general performance of the chaotic waveform is shown to be better than DS, yet no physical communication system can completely prevent multipath degradation when spurious images occur within one chipping sequence duration. Practical alternatives for distortion mitigation lead to the development of a RAKE receiver in addition to a novel combination of chaotic/CAZAC signals with binary offset coding (BOC) modulation.

5.1 Transmission Channel Model Effects

The effects of physically transmitting the communications signal through a live channel requires some understanding of the natural distortions and mitigation techniques that exist. A qualitative summary of the distortions and modeling implications are described below:

► Especially at lower frequencies, the noise floor is not flat, making the SNR of a spread spectrum waveform frequency dependent. Likewise, pathloss that comes from transmitting the signal is frequency dependent with higher frequency signals yielding higher pathlosses. When known or measurable differences in frequency selective channel performance are experienced, one mitigation technique is to predistort the spectral content of the flat chaotic waveform such that channel capacity communications are realized.

► Multipath images occur due to multiple signal paths between the transmitter and receiver. Especially in mobile communications where the signal path changes rapidly, multipath

can result in constructive or destructive interference at the receive antenna, boosting or eliminating the received signal power. Spread spectrum communication systems provide a natural resistance to multipath interference since the images must occur at the higher chip rate, reducing the delay spread of concern. Various methods to reduce multipath effects include receive antenna diversity, directional antennas, interlaced modulations, and equalization.

► Fading channel characteristics may produce flat, dispersive, and even nonlinear distortions to the waveform that perturb the received signal, resulting in reception dropouts or a loss of reception altogether. The most practical method used to mitigate reception dropouts is forward error correction or other higher-level protocol functions that ensure data integrity.

► Channel equalization routines are commonly used to combat the effects of channel distortions by evaluating the received characteristics of a known data sequence and applying correction factors to a transmission. One such example is the acquisition and training preamble that is prepended to wireless data packets in 802.11 protocols.

► Similarly, time-varying distortions caused by signal transmission through the Earth's ionosphere and troposphere limit the pseudoranging capability of a geodetically fixed receiver. Chaotic waveform variants, including noncoherent waveforms on multiple carriers, offer the potential for improved signal reception via spreading chip level resolutions in ionospheric/tropospheric corrections, aided by the non-periodicity (maximal entropy) of the waveform. It is also believed that the chaotic waveform variants will yield better performance in fast-varying scintillation environments.

► The impulsive autocorrelation of the chaotic waveform leads to a potentially strong RAKE receiver performance, taking advantage of delayed multipath images to receive a transmitted signal in lieu of a significantly distorted specular component.

5.1.1 Multipath Characteristics

Spread spectrum communication systems rely on the spreading sequence to first spread and then despread a modulated data stream to transmit data through a communications channel. The desire is to have a perfectly impulsive autocorrelation function in order to minimize the impacts of multipath images; traditional DS spread communication systems improve the multipath performance by spreading the signal, yet they are also susceptible to noncentral autocorrelation peaks and cyclostationary effects that decrease with spreading sequence length. Even infinite length DS spreading sequences have finite autocorrelations that impact multipath performance, leading to other modulation methods like binary offset coding (BOC)[166] that rely on Doppler or other corrections to better resolve multipath images. As an example, consider the aperiodic auto-covariances of three basic DS spreading sequences[13]

as shown in Figure 97; on the left is a length-11 Barker sequence, in the center is a length-15 m-sequence, and on the right is a length-1023 GPS civilian acquisition code.

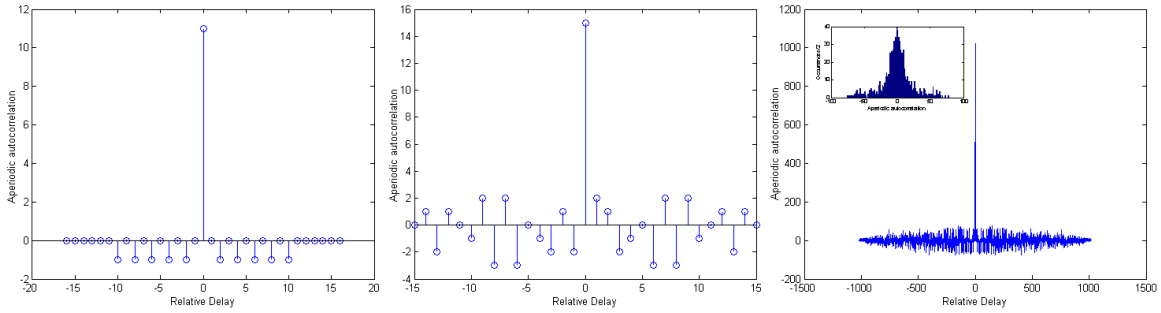

FIGURE 97. Aperiodic autocorrelation of DS spreading sequences: length-11 Barker code (left), length-15 m-sequence (center), and length-1023 GPS C/A code (right).

The multipath characteristics of a periodic spreading code result in additional signal images that are attenuated by the autocorrelation, leading to 10 additional terms attenuated by $\frac{1}{11}$ for the length-11 Barker code; 10 additional terms attenuated by $\frac{1}{15}$, 10 additional terms attenuated by $\frac{2}{15}$, and 4 additional terms attenuated by $\frac{3}{15}$ for the length-15 m-sequence; and a cumulative $\frac{15423}{1023}$ in potential images over the length-1023 GPS C/A code. The longer code lengths do significantly better than the shorter code lengths since the number of multipath images incident at an antenna are typically small.[48] A first-order measure of the expected multipath interference caused by the periodicity of a spreading sequence is an RMS measure of the energy over a small range of delayed signals ($\pm$100 chips), accounting for the autocorrelation effects, but ignoring path length difference attenuation. For practical comparison, values of the simulated chaotic waveform were autocorrelated over a range of practical offsets ($\pm$100000 chips) and compared to relatively long-length GPS spreading codes with cyclic repetitions of 1023 and $6.19 \cdot 10^{12}$ spreading chips[103], assuming identical spreading sequence rates. A plot of the aperiodic autocorrelation for each of these spreading sequences is shown in Figure 98.

[48]The length-1023 GPS C/A code lasts 1 ms, which is equivalent to path length differences of 300 km. Any multipath image that travels an additional path length of any fraction of 300 km will be attenuated out of consideration versus the specular signal component of a multipath equation.

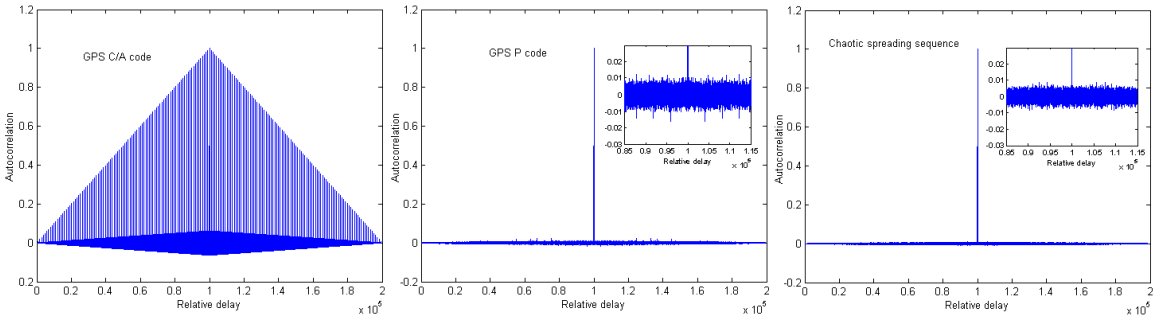

FIGURE 98. Aperiodic autocorrelation of length-100000 GPS and chaotic spreading sequences: C/A code (left), P code (center), and chaotic (right), including excised correlation peaks.

The shorter C/A code repetition period results in numerous correlation peak images, while both the P-code and chaotic sequence appear to have relatively impulsive autocorrelations. The closer in views of the P-code and chaotic sequence autocorrelation show that the average autocorrelation value is less for the chaotic sequence than for the effectively infinite DS P-code sequence. Quantifying this difference, which is largely due to the additional amplitude levels in the chaotic sequence, a plot of the expected RMS autocorrelation values over a range of relative delays on either side of the excised correlation peak is shown in Figure 99.

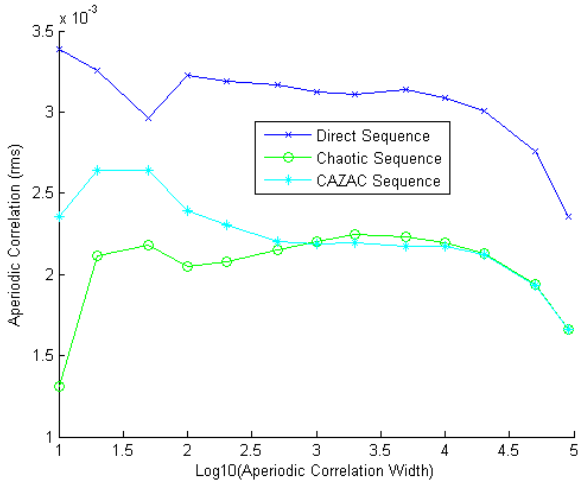

FIGURE 99. Aperiodic autocorrelation comparison for DS and chaotic spreading sequences.

The expected RMS value of the chaotic sequence standard deviation is approximately $\frac{1}{\sqrt{2}}$ that of the DS spreading sequence due to the chaotic signal being complex valued while the DS signal is real valued; use of a quadrature DS spread waveform yield similar long-term correlation results, indicating similar multipath and co-interference characteristics. Seperability of the chaotic signal is marginally better than for DS signals[72], suggesting similarly marginal

improvement in multipath performance.

5.1.2 Fading Channels

Fading in transmission channels is a natural phenomenon where the multipath images cause a reception dropout that usually depends on the specific set of signal paths between transmitter and receiver. Flat fading is modeled as yielding equal degradation over the entire signal spectrum, while dispersive fading induces frequency varying characteristics to the fade. Flat fading approximations give good first-order indications of a communication system's fading performance. In particular, "many radio channels can be accurately modeled as [wide sense stationary uncorrelated scattering] channels," simplifying the analysis through the assumptions that multipath images have uncorrelated time delays, phase shifts, and attenuations[13]. Some frequency bands, such as the 2.4 GHz ISM band chosen for the prototype chaotic communication system do exhibit frequency selective fading. During testing of the prototype hardware chaotic communication system, dispersive fading effects were visually apparent at times, but did not appreciably affect reception; however, it was validated that the specular component had the greatest strength, making selection of the first correlation peak from the adaptive correlator the optimal one.

Noting from previous analysis that the chaotic waveform is indistinguishable from flat AWGN in any bandlimited spectrum, the inverse Fourier transform indicates a $\frac{\sin(x)}{x}$ characteristic; for spread spectrum systems with large signal bandwidths, this may be approximated as the inverse Fourier transform of a constant, which is simply an impulse function $\delta(x)$. In terms of multipath characteristics, the chaotic waveform will have a zero expectation at any $|\tau| > T_c$, consistent with the recognition of the impulsive autocorrelation. Similarly, the coherence time of the chaotic signal will be identical to the spreading chip duration, or $T_c = 100$ ns. Typical delay spreads of typical macrocellular communication systems are on the order of 1 to 10 μs, while indoor microcellular propagation may range from 30 to 300 ns[13]; the chaotic spread waveform, with a spread bandwidth of 1 MHz or greater, should see very limited multipath effects in macrocellular system. To characterize the closer in multipath performance, simulations were performed on comparable QPSK modulated DS and chaotically spread communication systems having delay spreads of $|\tau| < 10T_c$, yielding almost identical results.

During testing of the prototype hardware chaotic communication system, effects of channel fading were induced. The primary mitigation technique for transmission channel fades is the ability to inject baseband pilot carrier symbols[155] to facilitate re-acquisition or equalization of the transmitted signal. Acquisition and synchronization performance of the prototype chaotic communications system was extremely robust at spread SNR levels of greater than -10

dB. At noise levels below -13 dB spread SNR, synchronization performance was noticeably degraded due to the frequency loop initialization that occurs in the frequency offset estimator at the conclusion of acquisition processing; fading effects did not show any evidence of impacting acquisition performance with the measurable limits of the hardware, but were evident when induced under laboratory conditions.

5.2 Chaotic Communications in Interference Channels

A second source of signal distortion that occurs in practical communications channels is interference by other emitters operating in the same shared frequency spectrum. In multiple access communication systems, such as those discussed in Chapter 8, the source of interference is often other users; for the chaotic waveform, co-interference may be modeled approximately as an accumulation of flat noise with AWGN spectrum since orthogonally spread maximal entropy carriers will have no higher likelihood of affecting reception processing than background noise. Spread spectrum communication systems as a whole provide resistance to natural and intentional interference since the despreading process using a maximal entropy chaotic sequence evenly spreads the interfering energy across the spread bandwidth. Independently of Shannon's proof that bandlimited AWGN-like signals are ideal for channel capacity communication: work by McEliece has shown that maximal entropy signals are optimal for communications channels with interferers and power contrained jamming[167, 168].

Interference mitigation in the prototype chaotic communication system starts with an IF SAW filter that provides sharp rejection to RF signals outside the band of interest. Bandpass filtering after the ADC reduces the out-of-band interferers further, with the despreader evenly spreading the in-band interferer energy throughout the output. OTA testing of the prototype chaotic communication system was performed in the presence of unintentional interference, with a representative spectral image shown in Figure 100; a receive spread signal strength of +10 dB was used to more clearly demonstrate signal and interference effects. Note that the SAW filter provides approximately 25 dB of out-of-band noise rejection, yet also produces a noticeable differential group delay to the in-band signal (tilt). A variety of narrowband and wideband interferers were present in the communications channel, validating the robustness of the chaotic communications signal in the presence of interferers.

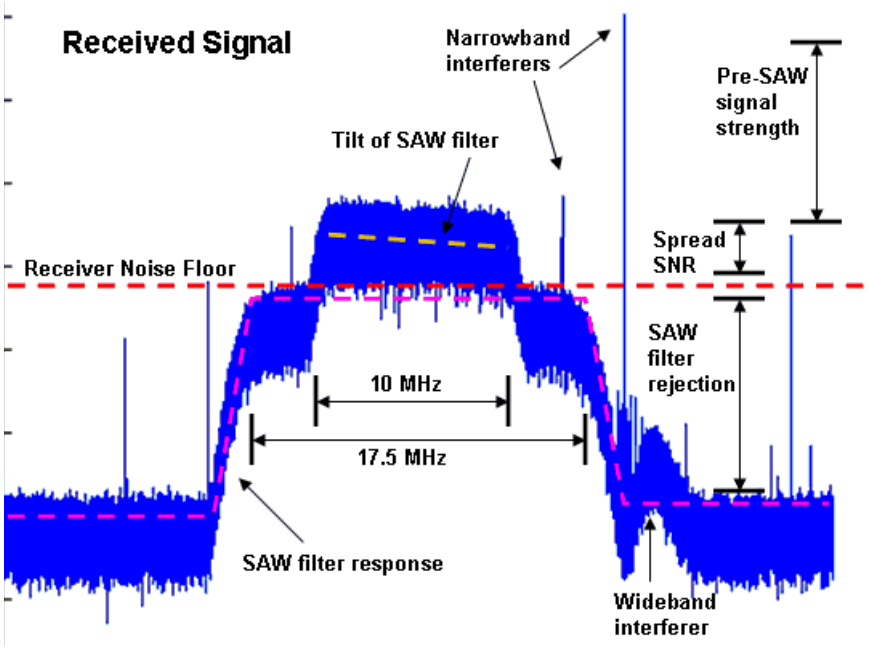

FIGURE 100. Chaotic signal receive spectrum in the presence of RF interference.

Performance of robust chaotic channel acquisition and demodulation was demonstrated using the prototype chaotic receiver without any significant impacts for line-of-sight transmissions. As both Shannon's[1] and McEliece's[167] analysis suggest, the performance of the chaotic signal in RF interference channels is primarily related to the signal vs. total integrated noise energy.

5.3 Chaotic RAKE Receiver

A common technique that is used in mobile communications to mitigate receive signal fading distortions is to implement a RAKE receiver, capable of locking onto and despreading distinct multipath images in independent reception "fingers" and then combining the despread signals. Adding this RAKE time diversity capability to the prototype chaotic receiver[169] may be implemented by employing additional despreader CMAC cells that combine the received signal and different delay/phase combinations of the internally generated chaotic sequence; tracking loops are implemented independently since the distinct multipath images represent different portions of the channel impulse response. Whenever one signal correlation is too distorted to provide a useable signal, the adaptive correlation can repeat its correlation search across potential delay/frequency offsets to locate additional multipath images. An exemplary block diagram of a coherent chaotic RAKE receiver is shown in Figure 101.

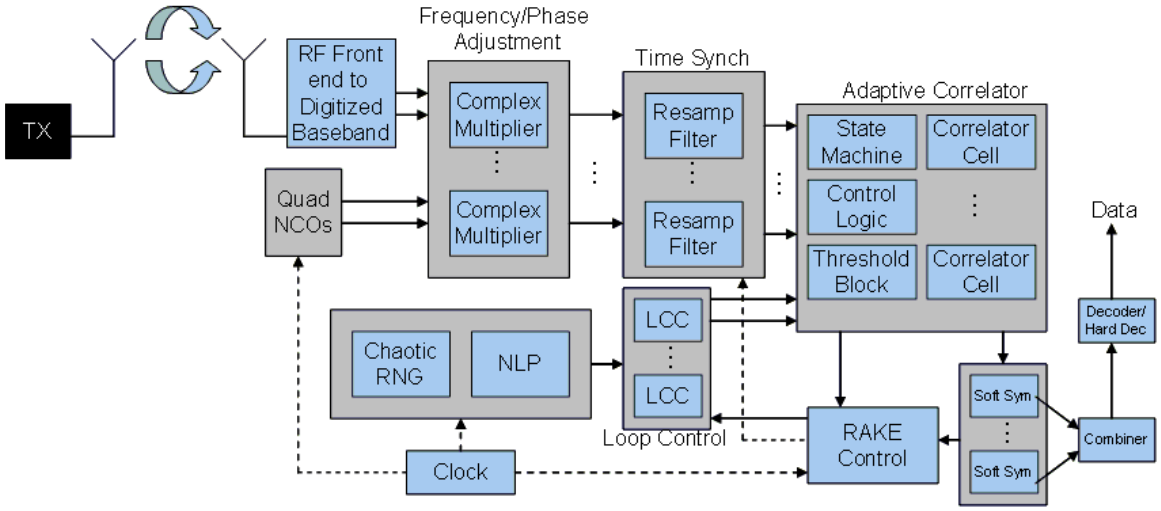

FIGURE 101. Exemplary block diagram of coherent chaotic RAKE receiver.

5.4 Binary Offset Coded Chaotic Waveform

A final mechanism that can be used to mitigate received signal distortions, and one that is particularly significant when the received signal characteristics are as important as the modulated data, is a binary offset coding (BOC) modulation that combines two or more subcarriers in an attempt to quantify channel distortions. A common example of BOC modulated spread spectrum communications is the GPS L1 Pseudo-M signal, where differential distortion characteristics of the two subcarriers give insight into correction factors for received signal arrival, Doppler rates, or atmospheric effects; the ultimate goal in reception of these GPS signals is to determine the pseudorange between a fixed point (GPS receiver) and a constellation of orbiting space vehicles. A measured Pseudo-M spectral image is shown in Figure 102; note that replacing the DS spreading sequences with chaotic sequences eliminates all sidelobe content. BOC modulation of orthogonal chaotically spread subcarriers provides improved multipath performance as was described in Chapter 5.1.1 in addition to the traditional BOC signal processing gains. Use of the CAZAC variant is likely preferred for space vehicle downlink transmissions and crosslinks.

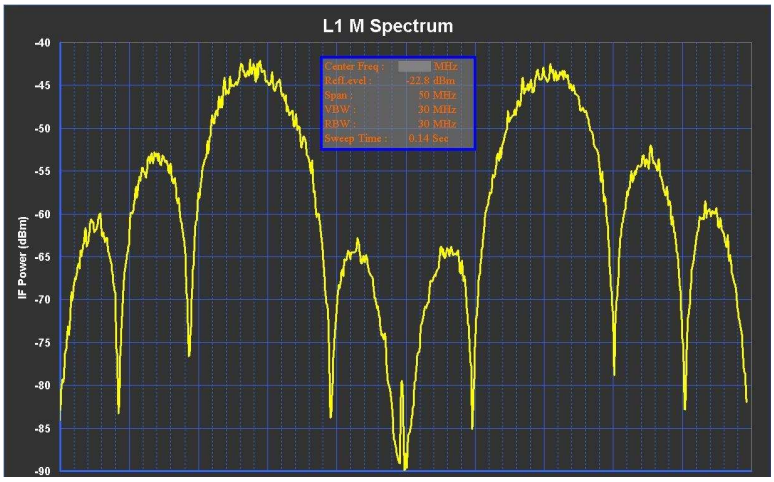

FIGURE 102. Exemplary binary offset coding (BOC) modulation characteristic based on GPS L1 Pseudo-M codes.

Chapter 6: PAPR-Adjusted Maximal Entropy Communications

The impulsive autocorrelation characteristic that lends superior multipath performance in the chaotic waveform is primarily a function of the uniformly random spreading sequence phase. At the same time, the chaotic waveform has a Rayleigh amplitude characteristic that results in relatively high peak to average power ratios (PAPRs) that are non-ideal in mobile communication systems. The coherency of the high PAPR signal does yield some advantages as presented with the selective noise cancellation technique, but those gains may not warrant the larger HPAs/RF circuitry. This chapter explores a novel modification of the basic chaotic waveform that parametrically reduces the amplitude modulation effects, yielding a low PAPR constant amplitude zero autocorrelation (CAZAC) waveform that is suitable for mobile communications, satellite ranging, and other power-constrained platforms. In addition, the parametric solution provides a simple mechanism for transitioning between low and high PAPR signals for environmentally responsive transmissions as would be found in a cognitive radio.

6.1 Generalized Chaotic Phase Shift Keying

The prototype coherent chaotic communication system presented in this dissertation implemented a straightforward chaos phase shift keying (CPSK) modulation, which is comparable in form to traditional digital PSK modulations. The quadrature CPSK may be expanded directly by increasing the number of data phases; resolving the data requires a higher symbol energy due to the reduced Hamming distance between data symbol constellation points. Expanding this technique to an extreme case that may be viewed as either a limiting case extension of DS spreading or an amplitude collapsed version of the chaotic waveform, a constant amplitude zero autocorrelation (CAZAC) spreading sequence is created. The CAZAC waveform is shown to have a significantly lower peak-to-average power ratio (PAPR) than the chaotic waveform in exchange for minimal cyclostationary features. Hybrid waveforms bridging the characteristics of the chaotic waveform and the CAZAC are constructed, with the ability to flexibly modulate PAPR and cyclostationary features based on intended use.

Noting that the chaotic spreading sequence is effectively a Rayleigh magnitude sequence times a quadrature sine/cosine sequence representing a uniformly random phase, it can be seen that adding any arbitrarily chosen and/or time-varying phase shift to a sequence (or symbol) will not induce cyclostationary features in the waveform. Provided the receiver has

a coherent duplicate of the chaotic sequence, this arbitrary phase shift can represent data as in the binary or quadrature CPSK methods discussed previously or any arbitrary M-ary PSK constellation. The easiest conceptual form for the PSK symbol to take is a classical root of unity; that is, each of the M-ary phase symbols is a solution to the complex valued equation (cyclotomic polynomial) $z^M = 1$. In the reduced cases, $M = 2$ gives solutions of $z = \pm 1$, while $M = 4$ gives $z = \{1, j, -1, -j\}$. An example of extending this chaotic sequence modulation is provided to demonstrate the generalized concept, followed by a brief discussion of the more general transmitter modifications necessary to combine the M-ary PSK symbol.

6.1.1 Exemplary 6-PSK Chaotic Modulation

As an example, consider the case $M = 6$, such that the solutions to $z^M = z^6 = 1$ are $z = \{\pm 1, e^{\pm j\frac{\pi}{3}}, e^{\pm j\frac{2\pi}{3}}\}$. The amplitude pair mappings for the I and Q channels corresponding to the solutions of z are $\{(\pm 1, 0), (\pm 0.5, \pm 0.866)\}$. These constellation points are equally spaced on a ring of constant amplitude as shown in Figure 103.

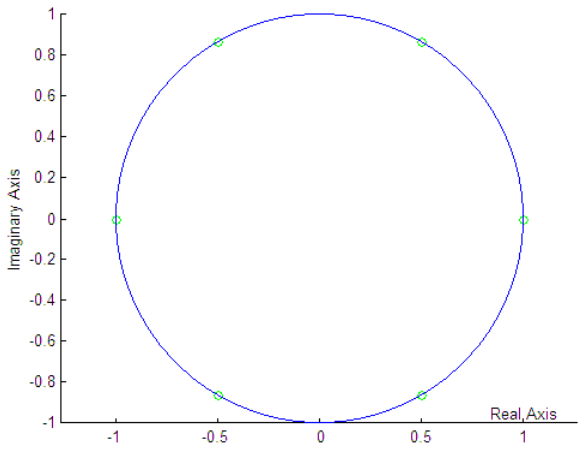

FIGURE 103. M-ary PSK constellation for M = 6.

The combination of these 6-ary PSK data symbols with the chaotic spreading sequence may be implemented using a straighforward complex multiplication; assuming the data symbol is transmitted with enough total energy to combat the reduced Hamming distance, the integrated symbols at the receiver may be reconstructed back into the original data symbols.

6.1.2 General M-ary PSK Modulations

In general, as the value of M increases, the constellation points become closer together (Hamming distance decreases), making the transmission more susceptible to errors in a noisy

channel. The more general case of M-ary PSK constellations yields constellation points at

$$z_n = \{e^{j\frac{2\pi}{M}n}\}_{n=0}^{M-1}$$

It is convenient to map these roots of unity to other analogous cyclic structures, creating a link between cyclic algebraic structures to the roots of unity. A block diagram of an exemplary M-ary PSK encoding and chaotic spreading sequence modulator is shown in Figure 104.

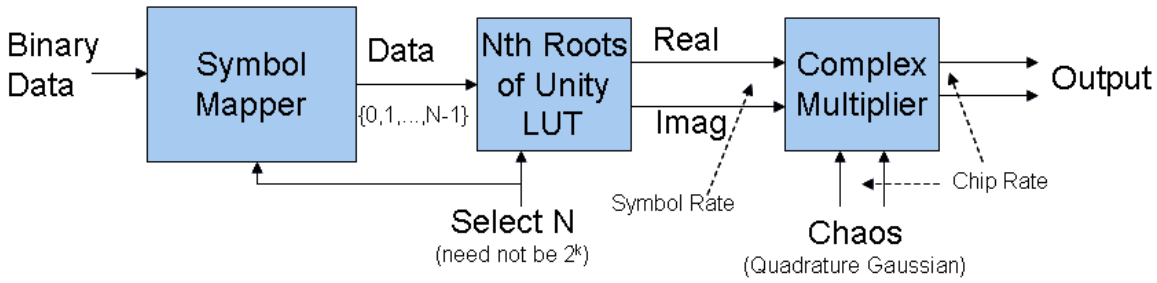

FIGURE 104. M-ary PSK chaotic sequence spread spectrum modulator.

The data that enters the symbol mapper is assumed to be in a binary format, but this can easily be modified for a mixed-radix system. The output of the Symbol Mapper may be viewed as either a complex-valued root of unity or an element in a finite ring. The latter may be preferred for a generic M-ary constellation to store the sine and cosine values in a zero-indexed lookup table that is accessed according to the ring elements passed to it by the Symbol Mapper. By design, these M-ary constellation points all lie on a circle of constant amplitude and induce an additional fixed angular shift in the chaotic spreading sequence for the duration of the symbol. All of the maximal entropy properties of the chaotic spreading sequence are maintained.

6.2 CAZAC Waveform

An interesting alternative to pseudorandom number generators constructed to yield maximal length sequences are those that have extremely small circular autocorrelations. These sequences are used in quantifying channel effects and equalizing receiver mechanisms based on the predicted channel response[170]. One set of sequences that has shown strong analytical properties in this area is constant-amplitude zero autocorrelation (CAZAC) sequences[171, 172], which are practically uncorrelated and can be used to compensate channel responses. A quick look at the properties of a CAZAC sequence shows that successive values have constant amplitude and uniformly random phase, which may be equivalent viewed as an M-ary direct sequence spreading waveform ($\lim_{M\to\infty}$) or as a reduction of the chaotic phase-shift keying waveform where the Box-Muller amplitude is collapsed to unity for all spreading chips. A

time-domain comparison of a chaotic phase-shift keyed waveform, a CAZAC waveform, and a DS spread waveform all based on the same uniformly random phases is shown in Figure 105.

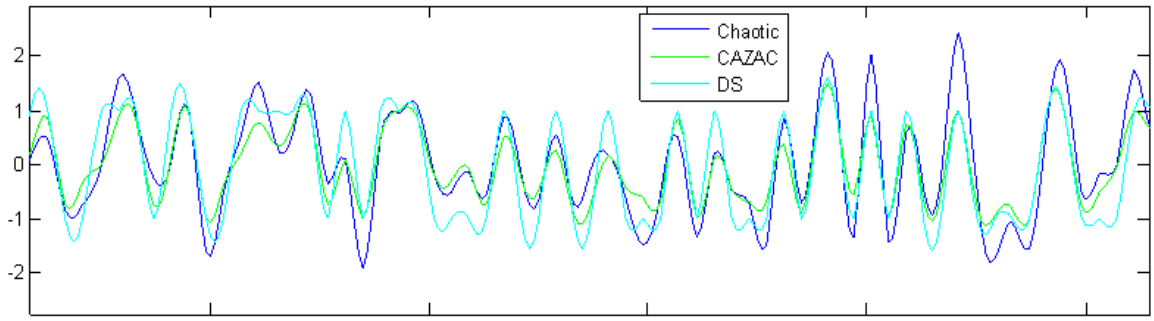

FIGURE 105. Comparison of chaotic PSK (blue), CAZAC (green), and DS (cyan) spread waveforms.

The few differences in implementation between the prototype chaotic PSK[160] and CAZAC[173] waveforms are achieved by bypassing the $\sqrt{-2\sigma_x^2 \log u_1}$ magnitude scaling in the Box-Muller transformation and adjusting the output signal level provided to the D/A converter to take advantage of the reduced PAPR. The simulated PAPR for the CAZAC waveform is approximately 3 dB as compared to practical values of 2 to 5 dB for quadrature DS spread waveforms. The receive processing is identical to a communication system using the chaotic spreading sequence, yet with a bypass of the symbol normalization and selective noise cancellation blocks. The CAZAC sequence is significantly different from the chaotic sequence in that the waveform is distinguishable from white noise; a phase-space plot of a CAZAC modulated signal (center) is compared to DS (right) and chaotic (left) in Figure 106.

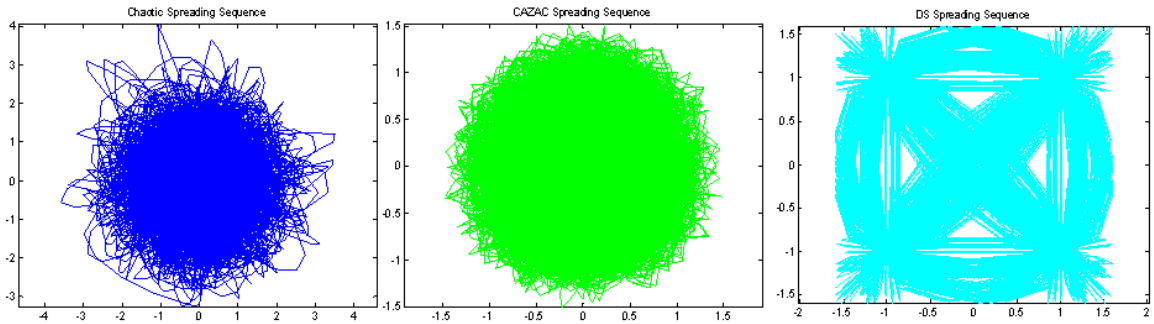

FIGURE 106. Phase space comparisons of chaotic (left), CAZAC (center), and direct sequence (right) spread waveforms.

The expected statistics for the CAZAC spreading sequence are a uniformly distributed amplitude in either the I or Q components[173], while simulations incorporating the effects of filtering cause slight deviations and peaked edge behavior similar to Gibbs phenomenon. A comparison of the CAZAC and chaotic waveform histograms is shown in Figure 107.

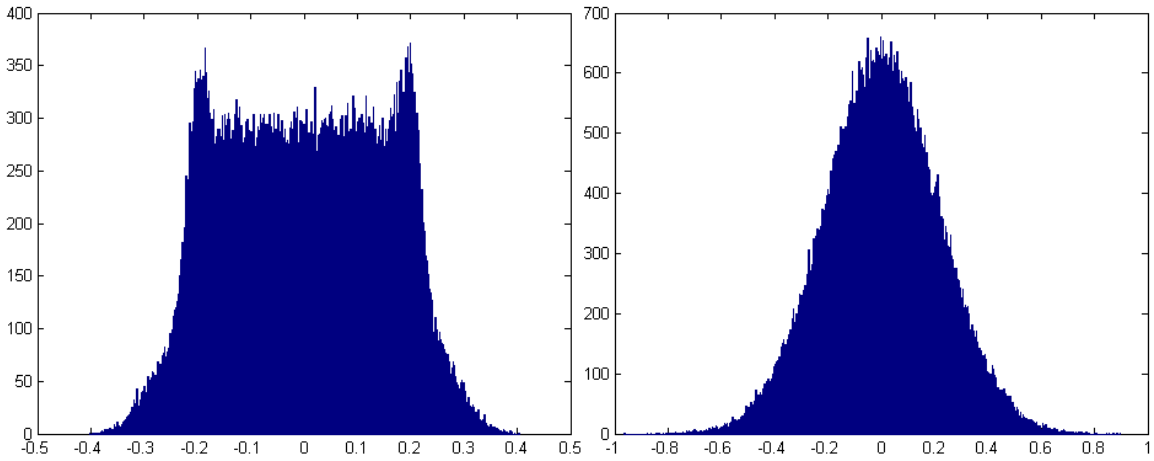

FIGURE 107. Histogram comparison of CAZAC (left) and chaotic (right) waveforms.

The frequency domain presence of the CAZAC waveform is expected to remain relatively flat, similar to the DS and chaotic spread waveform spectrums; a comparison showing similar results for all three is shown in Figure 108.

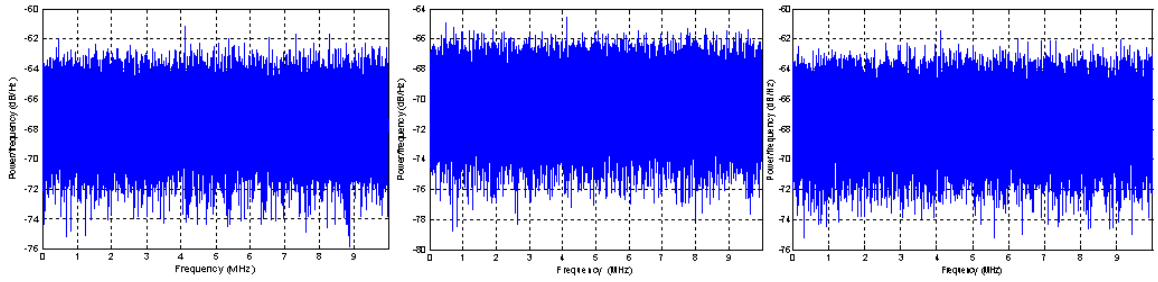

FIGURE 108. Frequency domain comparison of DS (left), CAZAC (center), and chaotic (right) waveforms.

A more detailed quantitative comparison of the CAZAC waveform requires evaluation of the higher-order statistics; a detailed evaluation of the first eight cumulants is captured in Table 9.[49]

[49]The simulation results are based on 1000 independent runs of the CAZAC waveform modulator in Matlab/Simulink, collecting samples covering 100,000 spreading sequence chips each.

TABLE 9. Cumulant evaluation of CAZAC modulated waveform

Cumulant	**Standard Normal**	**Uniform** $U(-\frac{1}{2}, \frac{1}{2})$	**CAZAC Sequence Mean**	**CAZAC Sequence Std Dev**
$1^{\text{st}}: \frac{\mu_x}{\sigma_x}$	0	0	$-4.45 \cdot 10^{-6}$	$5.17 \cdot 10^{-3}$
$2^{\text{nd}}: \frac{\mu_{x,2}}{\sigma_x^2}$	1	1	1	$2.39 \cdot 10^{-3}$
$3^{\text{rd}}: \frac{\mu_{x,3}}{\sigma_x^3}$	0	0	$-7.14 \cdot 10^{-5}$	$6.61 \cdot 10^{-3}$
$4^{\text{th}}: \frac{\mu_{x,4}}{\sigma_x^4}$	3	1.80	1.8411	$3.02 \cdot 10^{-3}$
$5^{\text{th}}: \frac{\mu_{x,5}}{\sigma_x^5}$	0	0	$-4.02 \cdot 10^{-4}$	$1.67 \cdot 10^{-2}$
$6^{\text{th}}: \frac{\mu_{x,6}}{\sigma_x^6}$	15	3.86	4.2159	$1.68 \cdot 10^{-2}$
$7^{\text{th}}: \frac{\mu_{x,7}}{\sigma_x^7}$	0	0	$-1.56 \cdot 10^{-3}$	$5.34 \cdot 10^{-2}$
$8^{\text{th}}: \frac{\mu_{x,8}}{\sigma_x^8}$	105	9.00	11.0838	$9.12 \cdot 10^{-2}$

In summary, the CAZAC variant of the chaotic waveform offers a much lower PAPR option with signal characteristics that more closely appriximate an ideal uniform distribution yet retain many of the practical benefits of the continuous phase, maximal entropy chaotic waveform. In particular, the CAZAC waveform is believed better suited for power disadvantaged platforms (mobile handsets, unmanned vehicles). The next step is to build a parametric bridge between the CAZAC and chaotic waveform, yielding a family of PAPR-adjusted waveform variants that flexibly support multiple applications.

6.3 PAPR-Adjusted Chaotic Phase Shift Keying

Realizing that the hardware implementation of the coherent chaotic and CAZAC based communication systems will be nearly identical, a generalized hybrid waveform was constructed[174] that modulates its PAPR based on an environmental control. This PAPR-modulated zero autocorrelation waveform ranges between the CAZAC waveform (minimum PAPR) and the chaotic waveform (maximum PAPR), also varying in level of cyclostationary feature content. Methods to create additional waveforms with custom amplitude modulation characteristics (different statistical distributions, expanded range of amplitudes, etc) follows the present discussion.[50] Consider again the previous statement of the Box-Muller transformation that is used in creating the chaotic sequence.

$$X_I = \sqrt{-2\sigma_x^2 \log u_1}\cos(2\pi u_2) \qquad X_Q = \sqrt{-2\sigma_x^2 \log u_1}\sin(2\pi u_2)$$

By collapsing the Rayleigh magnitude term and normalizing the variance ($\sigma_x^2 \equiv 1$), the sequence values for X_I and X_Q may be reduced to unity, leading to the uniformly random

[50]One variant is taking $\gamma > 1$ and using the selective noise cancellation approach to concentrate transmitted signal energy in short seemingly random time division bursts. The approach also allows the creation of shaped noise for simulations or other applications.

phase distribution of the CAZAC waveform. Consider now a more generalized form of the Box Muller distribution,

$$X_{I,\gamma} = \sqrt{2h(\gamma)}\,(-\log u_1)^{\frac{\gamma}{2}}\cos(2\pi u_2) \qquad X_{Q,\gamma} = \sqrt{2h(\gamma)}\,(-\log u_1)^{\frac{\gamma}{2}}\sin(2\pi u_2)$$

where γ is an amplitude stretching factor ranging from $[0,1]$ and

$$h(\gamma) = 0.053159\gamma^4 + 0.056998\gamma^3 - 0.69207\gamma^2 + 0.58221\gamma + 0.99985 \approx \frac{1}{\int_0^1 (-\log u_1)^\gamma\, du_1}$$

is a variance normalization factor for all $\gamma \in [0,1]$. This generalized transformation provides a constant spreading sequence variance

$$\begin{aligned} Var(X_{I,\gamma} + jX_{Q,\gamma}) &= E[(X_{I,\gamma} + jX_{Q,\gamma}) \cdot (X_{I,\gamma} - jX_{Q,\gamma})] \\ &= E[(2h(\gamma)(-\log u_1)^\gamma)] \\ &= 2h(\gamma)\int_0^1 (-\log u_1)^\gamma\, du_1 \\ &\approx 2 \qquad\qquad \forall\,\gamma \in [0,1] \end{aligned}$$

and also provides a wide range of amplitude scalings to adjust to a specific desired PAPR. A depiction of the modulated amplitude scaling over the range of u_1 is shown in Figure 109.

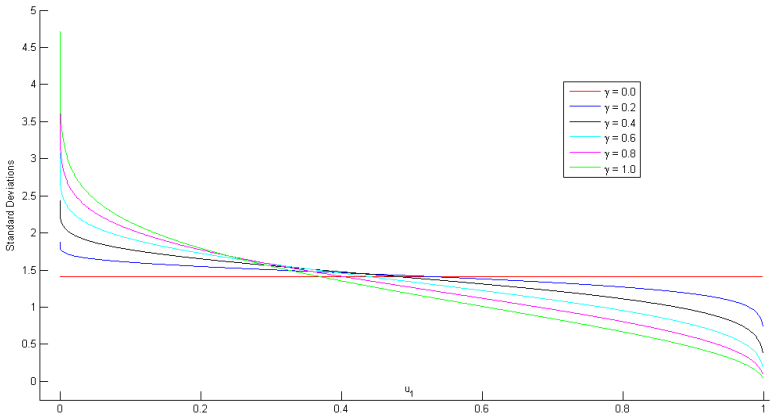

FIGURE 109. PAPR modulated waveform amplitude mapping.

To compare the tradeoffs between PAPR and cyclostationary features in the PAPR-modulated waveform, a simulation was performed for $\gamma \in [0,1]$ in 0.01 steps to obtain the predicted values. The curves in Figure 110 are depicted as fractional cumulant values relative to an ideal standard Normal distribution;[51] the wider the difference between the ideal values, the greater the cyclostationary feature content.

[51] An ideal standard Normal distribution has a variance of 1, a kurtosis of 3, a hexa-covariance of 15, and a PAPR of ∞; truncating the Normal distribution at $\sigma_{Trunc} \approx 4.67$ yields a practical PAPR of 13.39 dB.

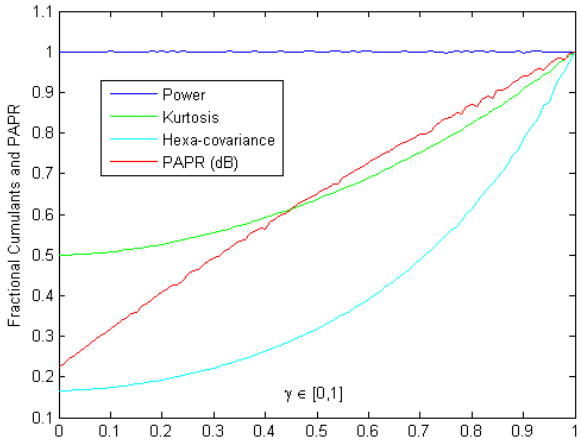

FIGURE 110. PAPR modulated waveform cyclostationary features.

Each of these curves was emperically approximated with a polynomial fit in order to provide a closed-form approximation of the waveform characteristics.

$$\begin{aligned}
\text{Variance}(\gamma) &= 1 \pm \epsilon \qquad |\epsilon| < 0.003 \\
\text{Kurtosis}(\gamma) &= 1.3491\gamma^2 + 0.14514\gamma + 1.4971 \\
\text{Hexa-covariance}(\gamma) &= 11.2104\gamma^4 - 10.281\gamma^3 + 11.7641\gamma^2 - 0.17253\gamma + 2.5112 \\
\text{PAPR}_{(\text{dB})}(\gamma) &= -2.0445\gamma^2 + 12.373\gamma + 3.0367
\end{aligned}$$

Using the empirical estimate of PAPR, the PAPR-modulated variant of the chaotic waveform provides approximately 10 dB of PAPR dynamic range[52] between the extreme cases of the CAZAC and chaotic waveforms. Taking the available ranges of PAPR that the PAPR-modulated chaotic hybrid waveform allows into account, it is possible to construct various communication systems with modes of operation that lie dormant until activated. A transmitter may have low average power emissions with mathematically proven absence of cyclostationary features (choosing $\gamma = 1$ for the chaotic waveform) and then burst higher rate or higher power transmissions once an environmental condition is activated. The simplest such condition is a rise in the intended receiver noise floor; such a raise may be caused by prevailing environmental conditions (e.g. lightning strikes in the HF spectrum or time of day) or an intentional jammer that is flooding the spectrum with targeted spurious energy. The use of the transmitter with a chaotic waveform, falling back into a degraded mode of a PAPR-modulated waveform, may be implemented without affecting the data rates. Moreover, a spot beam effect may be created for protected transmission of data like GPS signals. As a result, a flexible PAPR-modulated waveform that adjusts to its transmission environment may be implemented using γ as a commanded transmission parameter in the communications protocol in addition to dynamically

[52]The system level dynamic range does not fully reach 10 dB since the selective noise cancellation gains are similarly reduced by the reduced PAPR.

controlled spreading ratios.

6.4 Implementation of PAPR-Adjusted CPSK

Starting with the basic chaotic waveform implementation as a baseline, the PAPR-adjusted variants only require addition of the generalized Box Muller transformation and gain controls that adjust based on the chosen PAPR. The amount of resources used in the Rayleigh magnitude NLP can easily be duplicated to produce a set of fixed options or more simply create a subsidiary NLP tasked with approximating $\sqrt{2h(\gamma)}$. Recognition that the gain steps in any adaptable protocol algorithm will probably be finite tends to encourage the first approach.

6.5 Applications of Generalized CPSK Communications

The reduced PAPR CAZAC waveform and the generalized PAPR-adjusted chaotic hybrid provide a range of benefits in applications where PAPR impacts communications performance. In most communication systems, the trades begin with the data converters, followed by the data modulation characteristics, and then the DSP and RF circuitry needed to support waveform transmission. Described below are a number of potential applications, presented strictly at a notional level to preserve proprietary interests, that have been developed in conjunction with the prototype chaotic communication system or identified as areas of future research.

► **Cognitive radios** are an emerging concept whereby the physical layer communication mechanisms adapt to the transmission environment. The chaotic waveform provides good multipath performance and proven maximal entropy potential, while the PAPR-modulated variants provide 7-10 dB of additional signal dynamic range[174].

► **Precision timing and pseudoranging** applications include global navigation satellite systems (GPS, Galileo, GLONASS, WAAS) and communications through harsh transmission environments (plasma). The CAZAC waveform in particular offers a low PAPR solution for the power disadvantaged terminals, while the non-periodic and continuously random phase of the spreading sequence limits cycle slips, multipath errors, and improves pseudoranging performance. Combining the CAZAC spread waveform with a binary offset coding modulation provides a waveform with ranging performance that rival GPS precise positioning service.

► **Mobile Communications** includes a broad community of cellular users and unmanned platforms. Based on the low despreader implementation losses obtained with reduced precision despreading arithmetic, one particular implementation is to construct the uplink between a mobile handset (power disadvantaged) as a CAZAC modulated waveform and the downlink from the base station as a chaotic waveform.

Chapter 7: Amplitude Modulated Chaotic Communications

Continuing with generalizations of the basic chaotic waveform, another goal of a practical communication system is to efficiently increase the amount of data transmitted during any given time interval. One straightforward method to achieve higher data throughputs is to increase the symbol rate; this increase leads to correspondingly higher transmission bandwidths and energies for the same quality of service. Traditional digital communication systems achieve higher throughputs than comparable analog communication systems by employing higher capacity data modulations like M-ary PSK, M-ary QAM, 16APSK, etc. These higher order modulations do require higher transmission power levels to accurately receive the generalized data constellations, yet they do so in a spectrally efficient manner. This chapter explores an application of these higher order modulation schemes in chaotic communication systems, leading to a generalized approach for modulating arbitrary data constellations with a chaotic sequence, yet retaining all of the desireable characteristics of a maximal entropy waveform that is indistinguishable from AWGN.

7.1 Generalized Chaotic Amplitude Modulation

A key adjustment to higher data capacity digital communications is the use of modulated data constellations with multiple amplitude levels. The transmitter may be seen to encode data information into both the symbol amplitude as well as the symbol phase used in PSK modulations. The receiver is configured to demodulate the symbol, obtaining a soft symbol estimate that depends on both the signal phase and amplitude. Common amplitude modulated digital modulations include pulse amplitude modulation (PAM), quadrature amplitude modulation (QAM), and asymmetric phase shift keying (APSK); QAM constellations typically produce the best error rate performance in a linear channel, yet require additional HPA backoff as compared to APSK constellations, making the modulation choice application dependent. A practical downside of all amplitude modulations is that they are heteroskedastic – the average energy transmitted during each symbol interval changes independent of the spreading sequence. Returning this modulated symbol to a maximal entropy waveform where the expected symbol energy is constant forall symbols even though the modulated amplitude levels vary improves that performance; moreover, it ensures that the signal is indistinguishable from bandlimited AWGN without the telltale cyclostationary features of amplitude modulated waveforms. Combining the featureless coherent chaotic modulation process with an amplitude modulated data constellation provides a significant opportunity for increasing data throughput, yet requires additional steps to prevent the induction of cyclostatioary features inherent

to the amplitude modulation. This chapter begins with a naïve extension of a 16QAM data constellation modulated with the chaotic sequence to demonstrate the induced cyclostationary effects. It continues with a exemplary solution for the case of chaotic modulated 16QAM and concludes with a generalized method for performing featureless chaotic modulation of any arbitrary data constellation. A brief description of hardware modifications required to create the generalized featureless chaotic modulations is also presented.

7.2 Naïve Extension to Chaotic Modulated 16QAM

As an example, consider the notional time-domain output signal and histogram for a chaotic sequence directly modulated with a 16QAM data stream as shown in Figure 111. Visually, the time-domain signal has a heteroskedastic nature, varying the power envelope of the modulated waveform on a symbol-by-symbol basis, while the histogram appears to represent a slightly peakish Gaussian distribution.

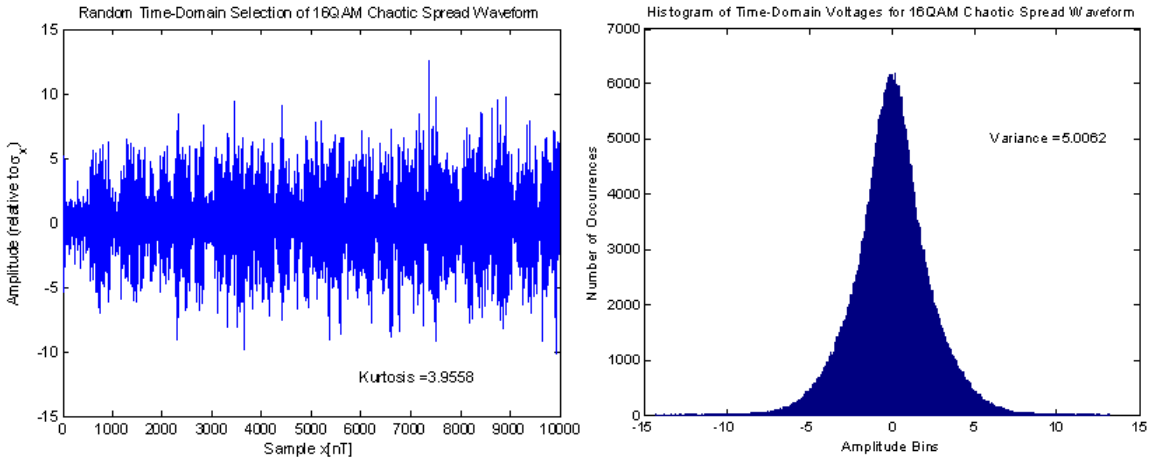

FIGURE 111. 16QAM symbols modulated with chaotic spreading sequence.

The expected kurtosis for this modulated waveform, which is a measure of its cyclostationary feature content, can be derived from a simple analysis of the 16QAM constellation points. Let σ_x^2 be the squared standard deviation (power) of the interior QAM constellation point. Then the second amplitude ring containing two constellation points is at a power of $\frac{10}{2}\sigma_x^2 = 5\sigma_x^2$. The outermost amplitude ring is at a power of $\frac{18}{2}\sigma_x^2 = 9\sigma_x^2$. The standard definition of kurtosis is the fourth central moment divided by the square of the variance; assuming that the data stream approximates *iid* uniform samples, the kurtosis computation is as follows.

$$\text{Kurtosis}_{16QAM} = \frac{\mu_{x+2\cdot(5x)+9x,4}}{\mu^2_{x+2\cdot(5x)+9x,2}} = \frac{\frac{1}{4}\left(\mu_{x,4} + 2\mu_{5x,4} + \mu_{9x,4}\right)}{\left[\frac{1}{4}\left(\mu_{x,2} + 2\mu_{5x,2} + \mu_{9x,2}\right)\right]^2} = \frac{\frac{1}{4}(3\mu^2_{x,2} + 150\mu^2_{x,2} + 243\mu^2_{x,2})}{\left[\frac{1}{4}\left(\mu^2_{x,2} + 10\mu^2_{x,2} + 9\mu^2_{x,2}\right)\right]^2} = \frac{99}{25} = 3.96$$

Note that the replacement $\mu_{x,4} = 3\mu_{x,2}^2$ is valid for any constant power envelope waveform modulated with the chaotic spreading sequence. Moreover, the mean is consistently zero for the chaotic sequence. The histogram of the 16QAM modulated waveform will still appear Gaussian, since the sum of Gaussian random variables with different variances is again Gaussian (in this case, the new expected variance is $\frac{1}{4}(1+5+5+9) = 5.0$). The waveform is distinguishable, however, by inspecting the time-domain waveform or calculating the higher order statistics like excess kurtosis. A second way to view the deviation from maximum entropy is to note that the transmitted signal is heteroskedastic, which results in cyclostationary features measurable with a delay and multiply envelope detector. Therefore, this naïve extension of directly modulating the chaotic spreading sequence with a 16QAM, or any amplitude modulated waveform, does induce entropy reducing cyclostationary waveform features that significantly move the combination away from Shannon's ideal channel capacity waveform. It does, however, point out that the induced deviations occur on a symbol-by-symbol basis since any one repeated symbol would be indistinguishable from AWGN.

7.3 Featureless Coherent Chaotic 16QAM

To implement chaotic spreading sequence modulation of an amplitude modulated waveform, there must be an instantaneous amplitude adjustment that does not affect the QAM constellations. One solution is to intentionally add a companion chaotic spreading sequence that is orthogonal to the first sequence, yet tightly coupled in symbol amplitude. This second sequence may also be used to encode redundant or additional information, making the net impact on the SNR performance minimal when spreading ratios are sufficiently large. The impulsive autocorrelation of the chaotic sequence ensures orthogonality and separability[72] of both the primary and complementary signals without appreciable energy loss. As a constructive example, consider the 16QAM constellation shown in Figure 112, which has three distinct power envelopes.

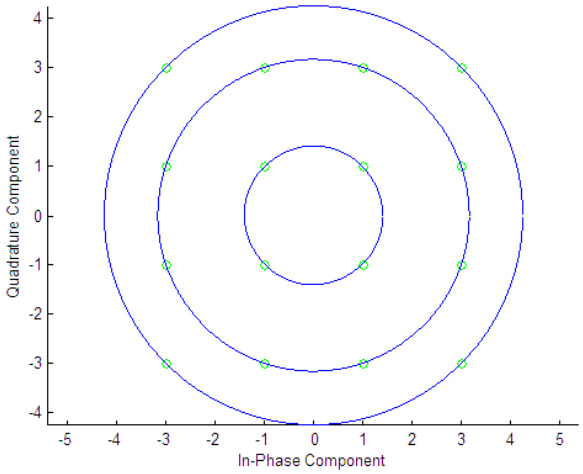

FIGURE 112. Constellation for 16QAM modulated symbols.

The unshaped PAPR of 16QAM is obtained from the ratio of the power envelope at the outermost constellation point to the average power envelope. Fixing the amplitude of the interior constellation point envelope power to x, we obtain the peak power envelope of $9x$ and the middle ring constellation point power envelope of $5x$. Therefore, the PAPR is

$$\frac{9x}{\frac{1}{4}(x + 2 \cdot 5x + 9x)} = \frac{9}{\frac{1}{4}(1 + 2 \cdot 5 + 9)} = 1.8 \approx 2.55 \text{ dB}$$

The time-domain waveform shown in Figure 111 gives an easy indication that the QAM modulated waveform has distinguishable features; those features are quantifiable as a standard kurtosis of 3.96, or an excess kurtosis of 0.96. Therefore, there must be a method that has a proven kurtosis (and all other cumulants) identical to that of a single Gaussian sequence if a chaotic modulated 16QAM data constellation can be constructed into a valid maximal entropy waveform.

Consider instead two orthogonal but otherwise identical chaotic spreading sequences, x_1 and x_2. Taking the first sequence x_1 to dynamically modulate a 16QAM unshaped data sequence, the waveform discussed in Section 7.1.1 is obtained with its inherent limitations. Since the two sequences are orthogonal, the power of the second sequence may be varied complementarily and added to the first sequence to help maintain a constant variance and mask the amplitude modulation effects. The statistical independence of the two orthogonal sequences make the average power of the sum equal to the sum of the powers, providing an easy avenue to eliminating the heteroskedastic output power.

$$\sigma^2_{x_1+x_2} = \sigma^2_{x_1} + \sigma^2_{x_2} + \rho_{1,2}\sigma_{x_1}\sigma_{x_2} = \sigma^2_{x_1} + \sigma^2_{x_2} + 0\sigma_{x_1}\sigma_{x_2} = \sigma^2_{x_1} + \sigma^2_{x_2}$$

By selecting the instantaneous power envelope of the second sequence equal to the difference of the (larger) constant power envelope and the instantaneous power envelope of the first signal, a constant power envelope with ideal even order cumulants is maintained. In the case of 16QAM, the three relative power levels are $\sigma_{x_1} \in \{\sqrt{2}, \sqrt{10}, \sqrt{18}\}$. The corresponding power levels for the second sequence are calculated as

$$\sigma_{x_2}^2 = \sigma_{x_1+x_2}^2 - \sigma_{x_1}^2$$

Choosing $\sigma_{x_1+x_2}^2 = 20$, the corresponding power levels are the inverse 16QAM mappings.

$$\text{Magnitude Mappings } (x_1, x_2): \quad \{(\sqrt{2}, \sqrt{18}), (\sqrt{10}, \sqrt{10}), (\sqrt{18}, \sqrt{2})\}$$

To demonstrate that the combined output retains the proper statistical properties, consider the kurtosis calculation discussed previously, modified for the two signals. Let x_1^* be the 16QAM modulated form of the chaotic spreading sequence x_1 and x_2^* be the "complement amplitude" 16QAM modulated form of the second chaotic spreading sequence x_2. Note that a linear combination of two unmodulated zero-mean statistically independent Normal sequences x_1 and x_2 of fixed power envelope still has a standard kurtosis of 3; therefore, the fourth central moment of the independent linear combination satisfies the following:

$$\begin{aligned}
\mu_{(\alpha x_1+\beta x_2),4} &= 3\left[Var(\alpha x_1 + \beta x_2)\right]^2 \\
&= 3\left[Var(\alpha x_1) + Var(\beta x_2)\right]^2 \\
&= 3\left[\alpha^2 Var(x_1) + \beta^2 Var(x_2)\right]^2 \\
&= 3\left[(\alpha^2+\beta^2)\sigma_x^2\right]^2 \\
&= 3\left(\alpha^2+\beta^2\right)^2 \sigma_x^4
\end{aligned}$$

The kurtosis of the combined chaotic spreading sequences modulated with chosen QAM amplitudes is identically 3 since the variance of the linear combination is $\left(\alpha^2+\beta^2\right)\sigma_x^2$. Repeating this calculation for the higher order cumulants also matches the analytical ideals for a normal distribution. As a result, the chaotic-QAM signal will have identical statistical properties and detectability as the single sequence chaotic spread PSK, but with an increased power by factor $(\alpha^2+\beta^2)$. For the specific case of 16QAM,

$$
\begin{aligned}
\text{Kurtosis}(x_1^* + x_2^*) &= \frac{\mu_{x_1+x_2,4}}{\mu_{x_1+x_2,2}^2} \\
&\approx \frac{\frac{1}{4}\left[\mu_{(x_1=\sqrt{2},x_2=\sqrt{18}),4} + 2\mu_{(x_1=\sqrt{10},x_2=\sqrt{10}),4} + \mu_{(x_1=\sqrt{18},x_2=\sqrt{2}),4}\right]}{\left(\frac{1}{4}\left[\mu_{(x_1=\sqrt{2},x_2=\sqrt{18}),2} + 2\mu_{(x_1=\sqrt{10},x_2=\sqrt{10}),2} + \mu_{(x_1=\sqrt{18},x_2=\sqrt{2}),2}\right]\right)^2} \\
&= \frac{\frac{1}{4}\left[3\cdot\left(\sqrt{2}^2+\sqrt{18}^2\right)^2\sigma_x^4 + 6\cdot\left(\sqrt{10}^2+\sqrt{10}^2\right)^2\sigma_x^4 + 3\cdot\left(\sqrt{18}^2+\sqrt{2}^2\right)^2\sigma_x^4\right]}{\left(\frac{1}{4}\left[\left(\sqrt{2}^2+\sqrt{18}^2\right)\sigma_x^2 + 2\left(\sqrt{10}^2+\sqrt{10}^2\right)\sigma_x^2 + \left(\sqrt{18}^2+\sqrt{2}^2\right)\sigma_x^2\right]\right)^2} \\
&= \frac{\frac{\sigma_x^4}{4}\left(3\cdot 20^2 + 6\cdot 20^2 + 3\cdot 20^2\right)}{\left[\frac{1}{4}\left(20\sigma_x^2 + 40\sigma_x^2 + 20\sigma_x^2\right)\right]^2} \\
&= \frac{1200\sigma_x^4}{\left(20\sigma_x^2\right)^2} \\
&= 3
\end{aligned}
$$

7.4 Generalized Featureless Chaotic Communications

Generalizing the preceding approach to a chaotic modulated 16QAM data constellation, the key realization is that the expected power envelope must be homoskedastic at all times, ensuring that the data sequence cannot affect the output signal characteristics absent a priori knowledge of the chaotic spreading sequence(s). The composition of complementary signals may be arbitrarily chosen with no interdependencies on phase modulation type, data rates, spreading ratios or protocol functions. In particular, a pre-selection of data types, user groups, access to information, data redundancy, data priorities, and other data specific characteristics may be built into the physical layer transmission while achieving near channel capacity data transfer.

Given N distinct data signals $\{s_1, s_2, \ldots, s_N\}$ that each represent amplitude and phase modulated data, the selection of signal content at any instant may be chosen arbitrarily given the constraints

$$\sum_{k=1}^{N} A_k^2 Var(s_k) = C$$

where A_k is the amplitude (notionally a voltage) of the amplitude modulated data symbol on signal s_k. Values of A_k may be binary for on-off keying, constrained to a constant for PSK modulations, or drawn from a finite set of amplitude rings associated with any digital modulation process. Noise performance of each signal is independent, provided the spreading gain on each s_k is sufficient to overcome the cumulative transmitted power C and the likely dominant receiver noise floor. Since the signals s_k are orthogonal and need not carry data at any commensurate rate, a filler signal may be employed to balance the emitted signals instantaneously, easing the signal constraints; this additional energy is wasted unless it is

associated with a signal or a coherent receiver.

$$\sum_{k=1}^{N} A_k^2 Var(s_k) \leq C \qquad \Longrightarrow \qquad A_{\text{fill}}^2 Var(s_{\text{fill}}) = C - \sum_{k=1}^{N} A_k^2 Var(s_k)$$

A simple example of this generalized featureless chaotic communications is a combined two signal ($N = 2$) chaotic modulated 16QAM and QPSK from the previous section, where 4 bits of 16 QAM data are encoded in the first sequence and 2 additional bits are encoded onto a QPSK constellation that is amplitude modulated complementarily to the instantaneous 16QAM symbol amplitude. Alternately, the last 2 bits of data may be encoded as a redundant phase angle to the instantaneous 16QAM symbol and complementary symbol amplitude; in this latter case, all energy transmitted is associated with the original 4 data bits, providing optimal noise performance via a QAM data constellation (optimal Hamming distances) with constant symbol energy for all symbols (all errors become equally likely). The superior separation capability of the chaotic sequence versus DS spreading approaches ensures low cumulative spreading losses.

7.5 Receiver Modifications for Chaotic Spread AM Constellations

Since the various chaotic spreading sequences are orthogonal, and moreover the expected value of $|\alpha| \ll 1$, they may be treated as entirely independent signals for reception purposes. In lieu of a complex multiplier used for despreading the received signal (or a trio of complex multipliers for early-late detection), the receiver re-uses its bank of complex multiply accumulator cells with distinct spreading codes to recover all signals simultaneously. The total received energy may be used for signal tracking purposes by combining the accumulated values in a coherent fashion. As is described in Chapter 8, access to the data carried signals s_k may be granted according to a selective permission-based scheme, further increasing the flexibility of data transmitted on a single carrier; unintended reception is sufficiently eliminated since the composite carrier provides an unbreakable noise floor to a partial permission receiver. Combining these observations results in a simplified receiver block diagram shown in Figure 113.

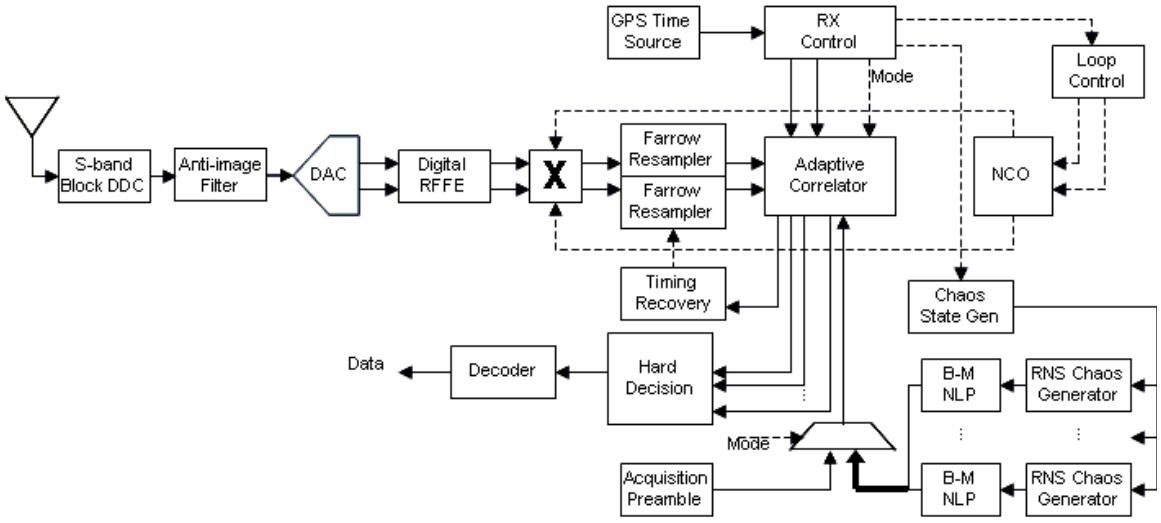

FIGURE 113. Coherent chaotic receiver for arbitrary data constellations.

The transmitter modifications are simpler, with filtering at the baseband sample frequency being shared instead of duplicated by the signals.

Chapter 8: Multiple Access Chaotic Communications

An immediate extension of the core chaotic waveform for larger-scale communication systems is multiple access communications. The DS-based code division multiple access (CDMA)[175, 176] standards invented for military and mobile communications have become the dominant solution for robust interferer resistant and multipath resistant communications modes in multiuser systems. The separability of the signals is achieved through the orthogonality of the DS spreading codes, permitting a broadly shared frequency spectrum with minimal interference to other users. CDMA technology has become well studied and accepted as one of the best communication modes for mobile users. DS systems do still contend with multipath degradations[177], multiuser co-interference[178], and the near-far transmit power control problem[179]. Most of these non-idealities are strictly due to the physics of wave propagation, making the practical goal mitigation of the effects. Chaotic sequence spread code division multiple access (CS-CDMA) communication systems using either the chaotic waveform or the CAZAC variant offer slightly improved performance by increasing the entropy of each signal in the channel, improving the separability or multiuser detection capabilities. Various authors[42, 180, 64, 181, 66, 182, 65] have identified the performance distinctions between DS-based and CS-based CDMA communications, with the general consensus being that chaotic communication system offer marginally better performance for multiuser detection[72, 183, 184, 185], better security and resistance to narrowband interferers[186], better multipath performance and applicability of RAKE reception[187], yet much greater susceptibilities to the nonidealities of a live communications channel[42, 64, 181]; the recognition that the mechanisms required for robust single-user chaotic communication systems have not been sufficiently developed leaves the optimization of those mechanisms for multiuser communications untouched in the open literature. Therefore, the purpose of this chapter is not to repeat the performance analysis of chaotically spread CDMA, but rather to present practical hardware and processing mechanisms for constructing robustly synchronized multiuser communications. In particular, three distinct derivations of general chaotic-based code division multiple access communication systems are discussed, built largely from the orthogonality properties of the chaotic/CAZAC spreading sequence, followed by three additional approaches to inherently secure chaotic multi-user communication systems. Optimization of these core concepts under various channel conditions is an area of continuing research.

8.1 Chaos-based Code Division Multiple Access Communications

The fundamental difference between single-user and multiuser communication systems is the method in which the signals are discriminated at the intended receivers, while all signals reuse a shared frequency spectrum. Detection and separation of the signals utilizes a time-synchronized replica of the spreading sequence to correlate and perform a threshold comparison. Chaotic signals show superior separation performance in linear AWGN channels[72], which is consistent with the indistinguishability of the chaotic waveform from background noise, making signals from other users appear to be increases in the flat noise floor. Moreover, the co-interference of multiple users in the same shared frequency spectrum is minimized since other signals appear to be noncoherent increases in the background noise level, conforming to McEliece's optimization of communication through interference[167]. This separation capability is ensured when the chaotic spreading sequences are coordinated such that no users transmit data spread with the same code in any defined geographic/temporal region. The core separation capability does not however address the traditional near-far power control problem. This section presents three distinct approaches to ensuring the orthogonality of the spreading sequences used in multiuser transmissions; a key component is the practical aperiodicity of the RNS-based digital chaotic circuit.

8.1.1 Basic Application of CS-CDMA

Chaos-based code division multiple access communication systems rely on the orthogonality of the spreading code, which presents a difficult mathematical problem unless an external reference or adaptive spreading code algorithm is used. The methods presented in this section[188, 189, 190] provide various mechanisms for allocating the spreading sequence(s) to multiple users, based on coordination of the sequence offset or initial condition, generation of independent chaotic sequences, and combinations of the two that segment both users and user groups. In general, any "good" PRNG[53] is believed to support the core of a chaotic spread communication system.

8.1.1.1 Shared Chaotic Sequences

The most straightforward method to extend the basic chaotic waveform is to assign users a unique identifier and, through that identifier, a personally owned portion of a defined chaotic sequence. Since the practical sequence lengths available to the RNS-based digital chaotic circuit are measured in googols, assigning 1 trillion distinct users (people or devices) a non-overlapping portion of the chaotic sequence leads to $\approx 10^{65}$ individual sequence lengths in the

[53] "Good" in the present context requires suitably infinite sequence length as well as efficient instantiation of initial states, state jumps, and PRNG processes.

assuming THz spreading ratios and 100 year leases of the sequence identifier. For all practical purposes, the number of distinct user IDs is infinite. To coordinate these IDs in the coherent chaotic communication system developed in Chapter 3, an additional free parameter that combines with the current GPS reference time defines the initial chaotic generator state at the beginning of transmission. That identifier or key is approximately 300 bits in length, consistent with the RNS-based representation of a one googol counter. The coherent chaotic receiver also applies the same key to its chaotic sequence generator, providing reception capabilities of the orthogonal code. To facilitate signal acquisition, higher level protocol functions may be implemented to either pre-coordinate the transmission start state/time or a generic preamble that provides the receiver with one-time knowledge of the transmitter's chaotic state (within the modulated data) to handover reception to the guaranteed orthogonal spreading sequence.

8.1.1.2 Independent Chaotic Sequences

The digital chaotic sequence generator discussed in Chapter 2 provides a number of flexible parameters for digital emulation of chaotic parameter modulation, changing the underlying chaotic evolution. More simply, choosing different chaotic polynomials, Galois Field characteristics, or even combination mechanisms provide clear approaches to ensuring orthogonality between multiple communications users. This approach translates into a different set of LUT coefficients and initial state parameters in the chaotic sequence generator, allowing flexible changes between sequences and underlying chaotic parameters without hardware modifications. The limitations in this approach are the finite number of small primes that lend themselves to chaotic polynomial computations (limited selection of RNS characteristics) and the additional programmability of the LUT values versus static RAM values (state information that must be transmitted to synchronize).

8.1.1.3 Coordinated Orthogonal Sequences

Combining these two approaches of assigning portions of the chaotic sequence according to a user ID and varying the underlying chaotic sequence parameters, a general chaotic sequence CDMA communication can easily be constructed. The first approach ensures orthogonality of distinct signals with a minimal amount of link layer overhead, while the latter approach permits flexible definition of user groups defined by chaotic parameters in addition to added security. An example of coordinate orthogonal sequences using intermediate output values of an identical chaotic sequence generator is shown in Figure 114. Note that the ring generator cells (RGC) are the smallest collection of CRT-combined ring generators that yield an approximate uniform distribution in $GF(2^{16})$, and the selection of intermediate result combinations all preserve the

uniformly distributed output. Further, note that this mixing structure at the output of the chaotic sequence generator, similar to a Feistal mixing structure in DES, provides simultaneous generation of loosely correlated sequences for wider band chaotic modulations.

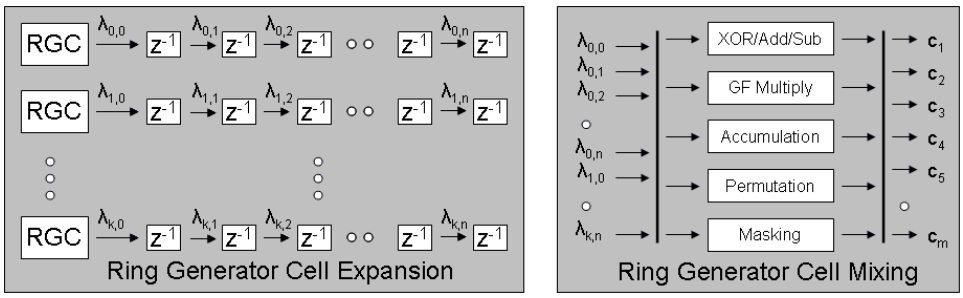

FIGURE 114. Exemplary chaotic sequence generation output mixing structure for coordinated orthogonal sequences from a single RNS-based circuit.

8.1.2 CS-CDMA Hardware Architectures

The hardware architectures of CS-CDMA transmitters and receivers are nearly identical to that of the prototype coherent chaotic communication system presented in Chapter 3. The fundamental difference is in the definition and control of the chaotic sequence state in the digital chaotic circuit. The transmitter architecture for transmitting a single signal is identical to the prototype, while a transmitter that simultaneously sends multiple orthogonal signals follows the construction of the maximal entropy chaotically spread amplitude modulation in Chapter 7. The key difference in hardware architectures is the multiple access receiver, such as would be used in a mobile communications base station. An exemplary architecture of a multiple access receiver capable of receiving independent chaotic signals is shown in Figure 115. The processing of each independent path is identical to the prototype chaotic communication system, taking advantage of the adaptive correlator flexibility by viewing the correlator cells as a resource pool that may be tasked to either acquisition or steady-state demodulation for any coordinated pair of received chaotic signal and time-synchronized internal replica.

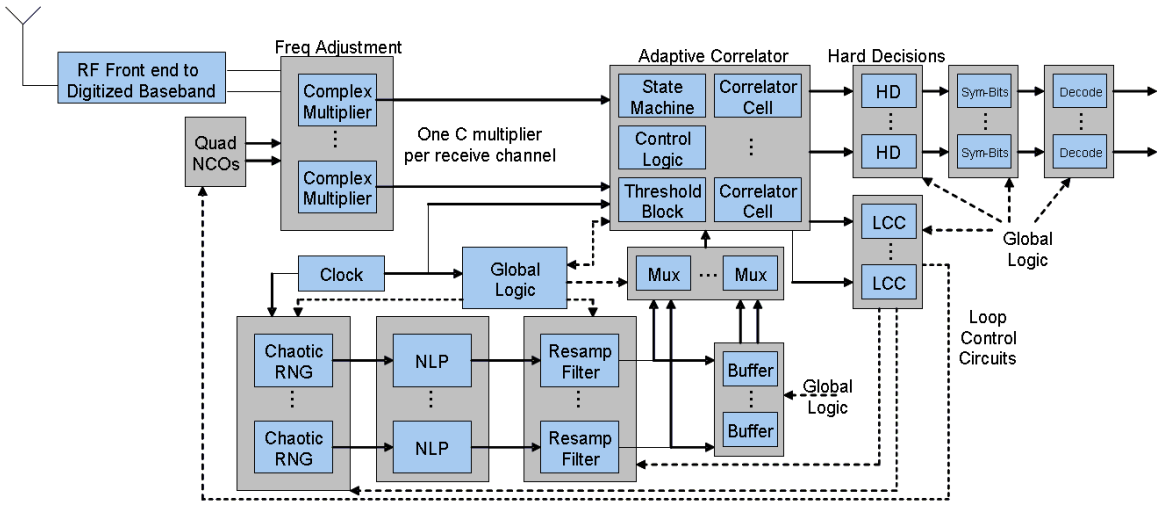

FIGURE 115. Chaotic sequence-based CDMA receiver architecture.

8.2 Secure Chaos-based Multiple Access Communications

Many large networks have a core requirement to disseminate information to multiple users or user groups while maintaining information security. Network protocols implement these security features in multiple ways, with the most obvious obfuscation method being encryption techniques that hide the data in plain sight; encryption increases the entropy in the data such that it cannot be efficiently understood without a series of keys or algorithms. Additional mechanisms include trying to hide the physical signal transmission from detection, but the physics of signal propagation make distance between a transmitter and a unintended receiver the greatest factor in detection algorithms. Chaotic signals provide some inherent security in that a maximal entropy waveform leaves little for an adversary to lock onto except power detection; methods to extend the use of spread spectrum techniques in secure communication system are well known in the published literature[73, 76, 36, 191, 133, 16, 41, 39]. This section presents three novel approaches to increased security in the physical layer modulation and transmission of a chaotic spread waveform.

8.2.1 Protected Amplitude Chaotic Communications

Returning to the generalized chaotic spreading approach for arbitrary data constellations, a mechanism exists to selectively encode data into a pair of modulated carriers, combined at digital baseband prior to transmission, such that permission based receivers have selective access to the transmitted information. More importantly, the protected data and global data signals are combined in such a fashion that all residual information content of the protected data is eliminated during the despreading process at a partial permission receiver. To construct

this protected amplitude chaotic communications mode, consider two otherwise identical orthogonal chaotic spreading sequences X_1 and X_2, each normalized to unit variance. Assume two independent data sequences, D_1 for protected data and D_2 for global data. The global data sequence D_2 is used to define an M-ary PSK symbol, $e^{j2\pi\frac{D_2}{M}}$ where $D_2 \in \{0, 1, \ldots, M-1\}$, that defines the phase rotation for the two independent chaotic spreading sequences. The protected data sequence D_1 is used to define a differing amplitude level between the two sequences; using the chaotically modulated 16QAM symbol that combines two independent chaotic carriers to construct a maximal entropy waveform as an example, the protected data sequence D_1 encodes a single bit of information as the amplitude of the first chaotic spreading sequence, corresponding to either the inner or outer 16QAM constellation amplitude level (voltage), $(1 + 2D_1)$. The amplitude of the second chaotic spreading sequence is chosen in a complementary fashion $(1 + 2(1 - D_1))$ to maintain constant total expected symbol energy. The two sequences are then added phase coherently to create a single signal composed of the two orthogonal carriers. A notional diagram of this modified chaotic transmitter processing is shown in Figure 116.

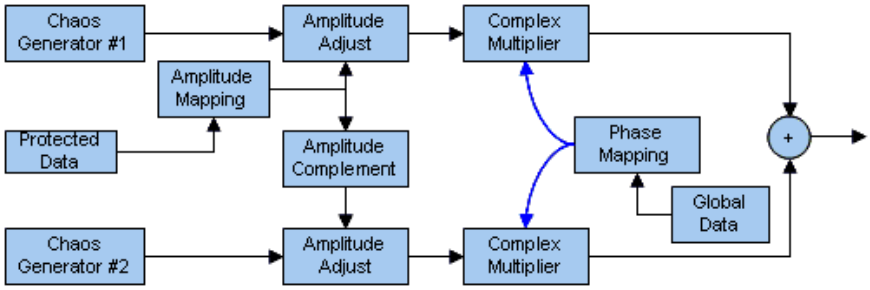

FIGURE 116. Protected amplitude data chaotic modulator.

Continuing the 16QAM example, the transmitted signal takes the form of

$$s_{Tx} = (1 + 2D_1)e^{j2\pi\frac{D_2}{M}}X_1 + [1 + 2(1 - D_1)]e^{j2\pi\frac{D_2}{M}}X_2 \qquad D_1 \in \{0, 1\}, \quad D_2 \in \{0, 1, \ldots, M-1\}$$

Two different receiver mechanisms are employed depending on the permissions of the user. In the partial permission case, the two independent chaotic spreading sequences are additively combined behind a tamper-proof boundary, resulting in a single chaotic sequence $(X_1 + X_2)$. The despreading operation on the received signal s_{Tx} eliminates the protected

amplitude information sequence D_1.

$$
\begin{aligned}
s_{Tx} \cdot (X_1 + X_2)^* &= \left((1+2D_1)e^{j2\pi\frac{D_2}{M}}X_1 + [1+2(1-D_1)]e^{j2\pi\frac{D_2}{M}}X_2\right) \cdot (X_1 + X_2)^* \\
&= e^{j2\pi\frac{D_2}{M}}\left((1+2D_1)X_1X_1^* + (1+2D_1)X_1X_2^* + [1+2(1-D_1)]X_2X_1^* + [1+2(1-D_1)]X_2X_2^*\right) \\
&= e^{j2\pi\frac{D_2}{M}}\left((1+2D_1)X_1X_1^* + [1+2(1-D_1)]X_2X_2^*\right) \\
&= e^{j2\pi\frac{D_2}{M}}\left((1+2D_1)E[X_1^2] + [1+2(1-D_1)]E[X_2^2]\right) \\
&= e^{j2\pi\frac{D_2}{M}}\left((1+2D_1)\sigma_{X_1}^2 + [1+2(1-D_1)]\sigma_{X_2}^2\right) \\
&= e^{j2\pi\frac{D_2}{M}}\left((1+2D_1) + [1+2(1-D_1)]\right) \\
&= 4e^{j2\pi\frac{D_2}{M}}
\end{aligned}
$$

In the full permission receiver, the two independent chaotic sequences are configured to despread the orthogonal transmitted carriers independently, leading to a phase decision $4e^{j2\pi\frac{D_2}{M}}$ that may be combined at baseband and an amplitude decision representing the protected data D_1. Despreading with internally generated chaotic sequence X_1,

$$
\begin{aligned}
Y_1 &= (s_{Tx} \cdot X_1^*) + (s_{Tx} \cdot X_2^*) \\
&= \left((1+2D_1)e^{j2\pi\frac{D_2}{M}}X_1 + [1+2(1-D_1)]e^{j2\pi\frac{D_2}{M}}X_2\right)X_1^* \\
&= (1+2D_1)e^{j2\pi\frac{D_2}{M}}X_1X_1^* + [1+2(1-D_1)]e^{j2\pi\frac{D_2}{M}}X_2X_1^* \\
&= (1+2D_1)e^{j2\pi\frac{D_2}{M}}\sigma_{X_1}^2 \\
&= (1+2D_1)e^{j2\pi\frac{D_2}{M}}
\end{aligned}
$$

A similar despreader correlation results in $Y_2 = [1+2(1-D_1)]e^{j2\pi\frac{D_2}{M}}$ for the X_2 despreading, permitting recovery of both the global data D_2 by adding the despreader results and the protected data by performing a threshold comparison between Y_1 and Y_2 to determine D_1. A comparison of the receiver processing for both the partial permission receiver (left) and the full-permission receiver (right) are shown in Figure 117.

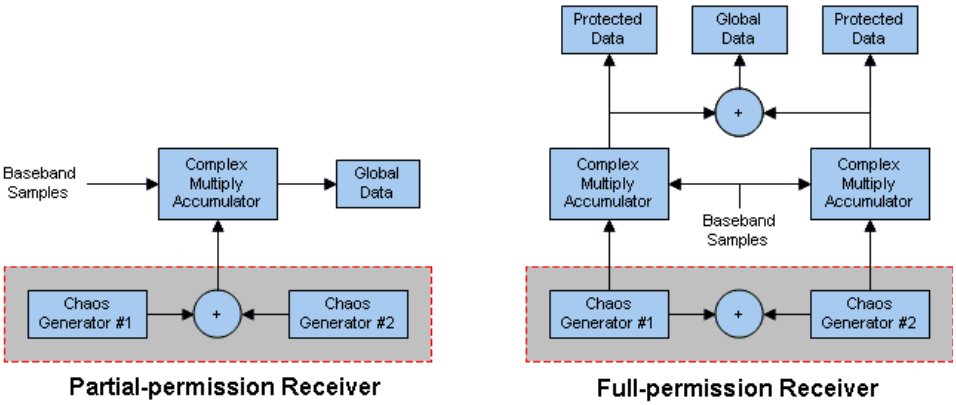

FIGURE 117. Protected amplitude data chaotic receiver.

This technique[192] may be extended to broader data constellations and even additional chaotic spreading sequences, but is limited by the geometric combinations of the despread symbols. A trivial extension of the technique that provides interesting results is combining the two chaotic sequences at the transmitter with an intentional $e^{j\pi}$ rotation between them and identical amplitudes; the partial permission receiver will obtain a correlation using its combined sequence $(X_1 + X_2)$ that has expectation zero, while the full permission receiver will receive two independent results that can be combined to a symbol with significant phase and amplitude information, correlating the input signal using $(X_1 - X_2)$.

8.2.2 Protected Phase Chaotic Communications

Similar to the protected amplitude chaotic communications that uses selective access to the chaotic spreading sequences to prevent access to protected data, a protected phase mechanism exists that provides another alternative for secure communications. Consider again two independent unit variance spreading sequences, X_1 and X_2. Assume again two independent data sequences, D_1 for protected data and D_2 for global data. The global data sequence D_2 is used to define an M-ary PSK symbol, $e^{j2\pi\frac{D_2}{M}}$ where $D_2 \in \{0, 1, \ldots, M-1\}$, that defines the bulk phase rotation for the two independent chaotic spreading sequences. The protected data sequence D_1 is now used to define a phase perturbation of the modulated symbols in steps up to one half the phase between global data phase steps, $e^{\frac{j\pi}{M}}$. The protected data D_1 is encoded in the phase shift between the X_1 sequence and a conjugate phase shift in the X_2 sequence. A block diagram of the transmitter modifications to modulate the modified chaotic constellation with protected phase information is shown in Figure 118.

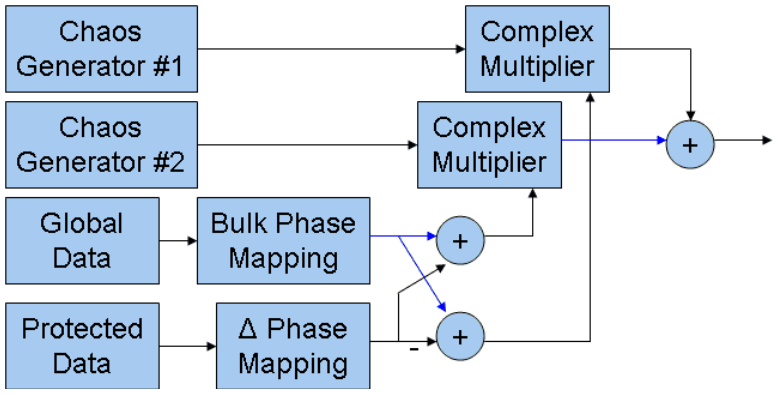

FIGURE 118. Protected phase data chaotic transmitter.

The transmitted signal takes the form

$$s_{Tx} = e^{j2\pi(\frac{D_2}{M}+\frac{D_1}{2M})}X_1 + e^{j2\pi(\frac{D_2}{M}-\frac{D_1}{2M})}X_2 \qquad D_1 \in \{-1,1\}, \quad D_2 \in \{0,1,\ldots,M-1\}$$

Separation of the protected and global data sequence is prevented at the partial permission receiver by again combining the two chaotic spreading sequences X_1 and X_2 behind a tamper-proof boundary, providing access only to the sum (X_1+X_2) at the despreader. The despreader output inside the partial permission receiver takes the form:

$$\begin{aligned} s_{Tx}\cdot(X_1+X_2)^* &= \left[e^{j2\pi(\frac{D_2}{M}+\frac{D_1}{2M})}X_1 + e^{j2\pi(\frac{D_2}{M}-\frac{D_1}{2M})}X_2\right]\cdot(X_1+X_2)^* \\ &= e^{j2\pi(\frac{D_2}{M}\pm\frac{|D_1|}{2M})} + e^{j2\pi(\frac{D_2}{M}\mp\frac{|D_1|}{2M})} \\ &= e^{j2\pi\frac{D_2}{M}}\left(e^{\pm j\pi\frac{|D_1|}{M}} + e^{\mp j\pi\frac{|D_1|}{M}}\right) \\ &= e^{j2\pi\frac{D_2}{M}}\left[2\cos(\pi\frac{|D_1|}{M})\right] \\ &= 2\cos(\frac{\pi}{M})e^{j2\pi\frac{D_2}{M}} \end{aligned}$$

Note that the choice of $D_1 \in \{-1,1\}$ and the even symmetry of the cosine function ensure that $2\cos(\pi\frac{D_1}{M}) = 2\cos(\pm\frac{\pi}{M}) = 2\cos(\frac{\pi}{M})$. The full permission receiver is able to discriminate between the marginal protected phase shift in each of the carriers, making a decision on the relative phase measurement since both transmitted signals are received with identical channel distortions. Despreading with internally generated chaotic sequence X_1 to obtain despreader

output Y_1,

$$\begin{aligned} Y_1 &= s_{Tx} \cdot X_1^* \\ &= \left[e^{j2\pi(\frac{D_2}{M}+\frac{D_1}{2M})} X_1 + e^{j2\pi(\frac{D_2}{M}-\frac{D_1}{2M})} X_2 \right] \cdot X_1^* \\ &= e^{j2\pi(\frac{D_2}{M}+\frac{D_1}{2M})} \\ &= e^{j2\pi\frac{D_2}{M}} e^{j\pi\frac{D_1}{M}} \end{aligned}$$

A similar despreader correlation results in $Y_2 = e^{j2\pi\frac{D_2}{M}} e^{-j\pi\frac{D_1}{M}}$ for the X_2 despreading, permitting recovery of both the global data D_2 by adding the despreader results and the protected data by performing a conjugate multiplication of the results $Y_1 Y_2^*$ and detecting the sign of the resulting phase.

$$\begin{aligned} Y_1 \cdot Y_2^* &= (e^{j2\pi\frac{D_2}{M}} e^{j\pi\frac{D_1}{M}}) \cdot (e^{j2\pi\frac{D_2}{M}} e^{-j\pi\frac{D_1}{M}})^* \\ &= (e^{j2\pi\frac{D_2}{M}} e^{j\pi\frac{D_1}{M}}) \cdot (e^{-j2\pi\frac{D_2}{M}} e^{+j\pi\frac{D_1}{M}}) \\ &= (e^{j2\pi\frac{D_2}{M}} e^{-j2\pi\frac{D_2}{M}}) \cdot (e^{j\pi\frac{D_1}{M}} e^{+j\pi\frac{D_1}{M}}) \\ &= e^{j2\pi\frac{D_1}{M}} \end{aligned}$$

Similar to the protected amplitude chaotic spread modulation, this technique[193] may also be extended to broader data constellations and even additional chaotic spreading sequences, but is again limited by the geometric combinations of the despread symbols that eliminate the protected data at the partial permission receiver.

8.2.3 Timeslotted Chaotic Communications

The previous two approaches have addressed modifications to the data modulation in order to separate users on the identical carrier, while another common approach is to use time diversity. Many wireless protocols use timeslotted bursts of distinct packets to distinguish information intended for each user. The packetized communications permit bounds on the channel distortions in the receive window, but also induce significant link layer overhead to maintain the network, especially in adhoc networking. An easy mechanism to create a chaotically spread time division multiple access communication system that has inherent security features is to divide the communications between users or user groups into distinct epochs and timeslots and adjust the chaotic sequence to a different state offset n_k during each epoch/timeslot. User groups with permission to access the information in any given timeslot have a time synchronized PRNG that aligns the initial chaotic states for acquisition processing at each burst, while users without permissions for access to information in a timeslot flywheel

through the timeslot or reacquire on the next permitted timeslot[194]; additional mechanisms to vary the chaotic state and sequence offset on timeslot boundaries include those stated in Chapter 8.1. A notional diagram of the transmitter modifications and the timeslots is shown in Figure 119. Additional techniques specific to directional TDMA networks have also been developed for neighbor discovery[195] and multi-tier chaotic communications[196].

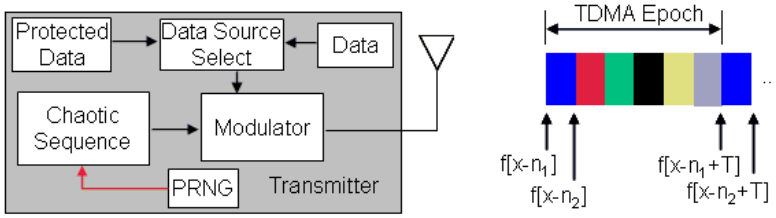

FIGURE 119. Time division multiple access chaotic communications.

8.3 Summary of Chaotic Multiple Access Communications

To summarize the basic extensions of the prototype chaotic spreading sequence to multiple access communications, there exists a clear, practicable mechanism for adjusting the digital chaotic sequence generator to allow separable multiple access communications approaching the performance suggested by the open literature[42, 64, 181]. The sequence generation modifications take the form of more detailed control in the chaotic sequence initial state, while the hardware modifications are mirror images of the multi-channel approaches to chaotically spread amplitude modulation waveforms. The permission based chaotic multiple access communication systems each use a key characteristic of the chaotic sequence spreading process and sequence orthogonality to simultaneously provide users or user groups with access to information based on physical layer network protocols; these methods are one level below the traditional encryption approaches, permitting a partial permission receiver to receive global data and have absolutely no access to protected data such as ciphertext. Such methods meet many of the objectives in secure communication systems in addition to the difficulty of an unintended receiver detecting and intelligibly deciphering a maximal entropy waveform. Future research will focus on optimizing the multiple access chaotic communications implementations and performance, hopefully making chaos a viable alternative to current 3G networks.

Chapter 9: Conclusions and Future Research

The research results presented in this document are believed to represent a practical basis for the implementation of coherent chaotic communication systems, constructed using an RNS-based digital chaotic circuit. This digital chaotic circuit was harnessed to analyze, simulate, and construct a hardware prototype of a simplex chaotic phase shift keying link operating at 2.4 GHz; both simulations and hardware measurements have validated the design and approach as a practically implementable solution to the robust synchronization of multiple chaotic circuits required for application in a chaotic communication system. This chaotic communication system, and the derived waveform variants, have the inherent maximal entropy characteristic described by Shannon, making them the ideal waveform(s) for channel capacity communications in a flat AWGN channel. Subsequent analysis and literature survey have shown the chaotic waveform to be superior to classical direct sequence approaches, both in the flat AWGN channel and in the presence of distortions. Extensions to that basic chaotic communications CPSK waveform were performed and presented as proven approaches for creating a broader class of chaotic-like waveforms with various advantages: the PAPR-adjusted variants of Chapter 6 permit extremely low PAPR transmissions (CAZAC waveform) while retaining the zero autocorrelation of the chaotic waveform, Chapter 7 provides a novel synthesis of the CPSK modulation characteristics leading to maximal entropy chaotic communication employing any arbitrary phase and amplitude modulated data constellation, and Chapter 8 discusses various approaches to standard and secure multiple access chaotic communications. The work presented in this dissertation is the subject of over 35 pending U.S. patents (Appendix B), with some areas of ongoing work not published in this document. Various additional techniques have been constructed for extending chaotic communication systems, many of which are areas of continuing research. A number of these techniques are provided in the following sections.

9.1 Dynamic Data Spreading Control in Chaotic Communications

A generalization of the constant symbol energy technique using a chaotic spreading sequence is the ability to dynamically control the data sequence characteristics relative to desired bit error rates, data throughput, or security considerations. Data throughput may be dynamically increased or reduced in a cognitive radio using algorithms that periodically evaluate channel conditions and subsequently vary transmit power and spreading ratios[173, 174]. Security in a chaotic spread communications can also be increased by dithering the symbol duration according to an independent random process such that an unintended receiver views

the symbol clock as a random walk[197]. Two additional dithering techniques have been developed at a chip level[198, 199], but are not included in the present discussion.

9.1.1 Environmentally Adaptive Chaotic Spreading Ratios

In addition to the PAPR modulation techniques presented in Chapter 6, a spread waveform lends itself to easy adjustment of the $\frac{E_s}{N_0}$ by varying the spreading ratio. In harsh environments, the data rate, and consequently the data throughput, may be reduced to the point that a satisfactory quality of service is obtained. Practical spreading ratios begin around 20 and extend as high as 200,000 for navigation data in GPS systems, providing 40 dB of dynamic range. Varying the spreading ratio within that dynamic range presents the possibility of an environmentally adaptive cognitive radio using a maximal entropy chaotic waveform. One simple method for constructing a practical system with this adaptive control is as a variation in the commanded E_{sym} as shown in Figure 120.

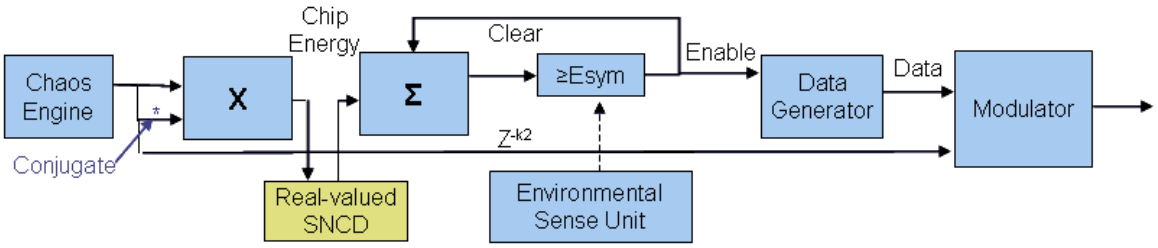

FIGURE 120. Environmental adjustment of spreading ratios.

9.1.2 Dithered Chaotic Symbol Durations

Extending directly on the constant symbol energy chaotic modulation, the symbol duration may be consciously dithered about the ideal by a pseudorandom process in order to further reduce the opportunity for an adversary to exploit. Given full access to the chaotic sequence, an adversary can lock onto the signal and begin demodulating the symbols. In this approach, the symbol boundaries are shifted randomly in time, causing a random walk in the symbol clock at an unsynchronized receiver. As a result, tracking operations become exceedingly difficult in the long run without knowledge of the pseudorandom symbol duration dithering process. A block diagram of this symbol duration dithering process is shown in Figure 121.

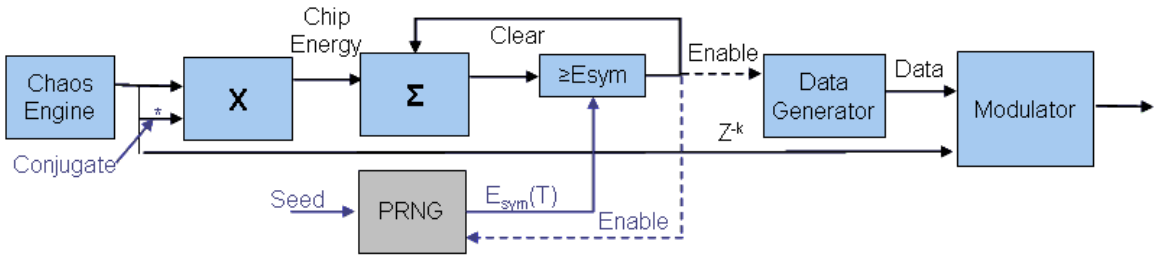

FIGURE 121. Symbol duration dithering mechanism.

9.2 Generalized Chaotic Carrier Modulation

The mathematical foundation of the higher-data capacity featureless chaotic amplitude modulation may be generalized by returning to the concept of linear combinations of orthonormal basis vectors. Given k independent (and therefore orthogonal) chaotic circuits during the n^{th} symbol period T_s that produce the orthogonal basis functions $g_{k,nT_s}(t)$, the transmitted symbol will take the form

$$y_{nT_s}(t) = \sum_k a_{k,nT_s} g_{k,nT_s}$$

where the complex-valued weighting coefficients a_{k,nT_s} represent the user data contained in the n^{th} symbol. The assumption of orthogonality forces the following for the chaotic sequences.

$$\int_{(n-1)T_s}^{nT_s} g_{k,nT_s}(t) g^*_{k,nT_s}(t) dt = C_{k,nT_s} \qquad \int_{(n-1)T_s}^{nT_s} g_{k_1,nT_s}(t) g^*_{k_2,nT_s}(t) dt = 0 \quad \forall k_1 \neq k_2$$

where C_{k,nT_s} is the energy of the k^{th} basis function (chaotic sequence) during the n^{th} symbol period. For ideal noise and maximal entropy channel capacity communications performance, $C_{k,nT_s} \equiv C_k = C$ is a normalized expected energy that remains constant during every symbol period. In reality, each sequence has a dynamic, yet apparently random, variance. The energy contained in the symbol is ideally equal to the sum of the orthogonal components, or

$$y_{nT_s}(t) \cdot y^*_{nT_s} = \sum_{k_1} \sum_{k_2} a_{k_1,nT_s} a^*_{k_2,nT_s} g_{k_1,nT_s} g^*_{k_2,nT_s} = \sum_k |a_{k,nT_s}|^2 |g_{k,nT_s}|^2$$

The ideal noise performance for this system occurs when all basis vectors $g_{k,nT_s}(t)$ have the same amount of energy per symbol period; that is, the set $g_k(t)$ form an orthonormal set during all symbol periods, even though the energy contained in any basis vector $g_k(t)$ will vary as a function of time. This energy may be shared among signaling components and maintained at a constant energy per symbol (with energy evenly shared among bits) through wise choice of weights a_{k,nT_s}.

Now consider the quadrature chaotic phase shift keying modulation used in the prototype coherent chaotic communication system. Two independent (orthogonal) chaotic sequences are combined using a complex multiplication with one of four symbols represented by a $\pm 1 \pm j$

data symbol. The energy of each chaotic sequence (basis vector) is spread evenly among the real and imaginary components, so that if a difference in energy between the two chaotic sequences is present during the symbol period, that energy will be statistically normalized to the in-phase and quadrature components at the output of the spreading multiplication. Stated in another fashion, the variation in symbol energy as detected in the observation vector at the coherent receiver will vary in magnitude according to the variance in the per-symbol energy, but strictly along the rays between the origin and the four ideal QPSK constellation points; the modulated energy is shared by either bit, making bit decisions from the observation vector sufficiently independent of the differences in energy by the chaotic sequences (time-varying basis functions). This two-dimensional case of chaotic QPSK modulation may be re-stated in terms of a two-dimensional unitary matrix multiplication as follows:[54]

$$y(t) = A \cdot \begin{bmatrix} g_1(t) \\ g_2(t) \end{bmatrix} = \begin{bmatrix} \cos(a_{nT_s}) & \cos(a_{nT_s}) \\ -\sin(a_{nT_s}) & \sin(a_{nT_s}) \end{bmatrix} \cdot \begin{bmatrix} g_1(t) \\ g_2(t) \end{bmatrix} \qquad a_{nT_s} \in \{-\frac{3\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}, \frac{3\pi}{4}\}$$

From an observation vector normalization perspective, the symbol energy as a whole, $|g_1(t)|^2+|g_2(t)|^2$, is normalized and then bit decisions are performed. Extending this quadrature modulation scheme to a third dimension, consider a collection of three independent chaotic sequence generators during any arbitrary symbol period

$$\{g_1(t), g_2(t), g_3(t)\}$$

that are combined to produce a symbol

$$y(t) = \sum_{k=1}^{3} a_k g_k(t)$$

The total energy contained within the symbol

$$\int_0^{T_s} |y(t)|^2 dt = \sum_{k=1}^{3} |a_k|^2 C_k$$

which will only result in optimal noise performance when each bit energy is $\frac{1}{3}$ of this total energy. The energy in each of the components $|a_k|^2$ may be viewed as the eigenvalue magnitudes of the orthogonal basis functions (consider the QR matrix decomposition[200]) and are identically equal to one for a unitary matrix A. The benefit of considering unitary matrices for the modulation process is that the demodulation process is both invertible and equal to the Hermetian adjoint of the modulation matrix[200]. A suitable unitary matrix modulation

[54]The scaling constant of $\sqrt{2}$ is simply an artifact of it being easier to multiply by $\pm 1 \pm j$ than $\pm\frac{\sqrt{2}}{2} \pm j\frac{\sqrt{2}}{2}$ in hardware; the scaling constant may equivalently be taken as linear gain outside of the modulation process.

process for a three-dimensional chaotic modulation is

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = A \cdot \begin{bmatrix} g_1(t) \\ g_2(t) \\ g_3(t) \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & -\frac{2}{3} \end{bmatrix} \begin{bmatrix} g_1(t) \\ g_2(t) \\ g_3(t) \end{bmatrix}$$

Note that the constant scaling factor of $\frac{1}{3}$ may be removed and compensated elsewhere in the transmitter as a gain term, reducing the modulation process to sums and differences of binary shifted chaotic sequences. The three-dimensional signal may be added and transmitted at baseband, but cannot be combined as in traditional communication systems where it is upconverted via orthogonal combinations of sine and cosine functions.[55] The signal may however be combined and "upconverted" using a second independent set of periodic orthonormal chaotic basis functions; these periodic basis functions may be implemented using a set of circularly addressed LUTs clocked at the desired upconversion rate. The resulting system has eliminated dependence on the periodic sine and cosine functions entirely, and may be increased to higher dimensions.[56] Further, conversion of the modulation process from a unitary matrix (data bits are encoded as plus or minus the eigenvectors) to a signaling set derived from constant-norm Normal matrices gives the ability to encode higher capacity data modulations in three, or higher, dimensions.

As an example, consider a three-dimensional extension of 16QAM, where 6 independent data bits are used to encode a three-dimensional symbol having constellation points at $\{-3, -1, 1, 3\}$ along each of the three orthogonal axes. The peak to average power is determined approximately from the constellation points, which may be broken down into any of the constellation octants. Thinking of a Hamming l^2 distance, the occurrence of inner-sphere (circle) constellation points having distance $\sqrt{3}$ from the origin is $\binom{3}{0} = 1$, constellation points having distance $\sqrt{11}$ from the origin is $\binom{3}{1} = 3$, constellation points having distance $\sqrt{19}$ from the origin is $\binom{3}{2} = 3$, and constellation points having distance $\sqrt{27}$ from the origin is $\binom{3}{3} = 1$. The ratio of peak to average distance from the origin is

$$\frac{\sqrt{27}}{\frac{1}{8}\left(\binom{3}{0}\sqrt{3} + \binom{3}{1}\sqrt{11} + \binom{3}{2}\sqrt{19} + \binom{3}{3}\sqrt{27}\right)} = \frac{24\sqrt{3}}{\sqrt{3} + 3\sqrt{11} + 3\sqrt{19} + \sqrt{27}} \approx 1.3877$$

This peak-to-average constellation point ratio can be converted to a PAPR estimate (actually a minimum) of $20 \log_{10} 1.3877 = 2.846$ dB, which is comparable to two-dimensional

[55]The completeness of l^2, and/or the ability to uniquely construct any time domain function using Fourier sequences of sines and cosines, ensures that there cannot be a third orthogonal periodic function to a sine and cosine at the same frequency.

[56]Further study of the effects that filtering and other components that are derived assuming sine/cosine periodicity is required to quantify this system and performance.

16QAM constellations at 2.55 dB. This three-dimensional 16QAM equally splits one-third of the signal energy into each of the orthogonal axes, leading to throughputs (assuming separability) comparable to three times a four constellation PAM signal with $10 \log_{10} \frac{1}{3} = -5$ dB reduction in signaling power. Specific applications calling for unique modulation characteristics may benefit from this higher-dimension modulation technique, as well as giving additional options for encapsulation of protected data in multiple access communications.[57]

[57]One example is a three-dimensional binary BPSK signaling constellation modulated using three orthogonal chaotic spreading sequences. A full permission receiver will have access to all three chaotic spreading sequences while a partial permission receiver has only one or two of the three sequences, leading to a three-dimensional despreader output at the full permission receiver and a one- or two-dimensional orthogonal projection for the partial permission receiver.

Appendix A: Initialization and Analysis Scripts

This appendix presents a collection of exemplary initialization and analysis scripts that provide simple results comparable to the Simulink/Synplify DSP models used to construct the prototype chaotic communication system.

► Script A.1 outlines the Rayleigh magnitude NLP initialization for use in the Box Muller transformation. The combined NLP implementation described in Section 2.3.3.5 results in approximately 14.5 bits of effective amplitude resolution.

► Script A.2 presents the complete initialization of the digital chaotic circuit used in the prototype chaotic communications system, including the non-chaotic masking generator. Values for all chaotic sequence generation processes begin with computation of irreducible chaotic polynomials $x^3 + 3x^2 + 3x + C$ over Galois Fields of characteristic p. Lookup table values and multiplication factors for the CRT combination are calculated and inserted into the Synplify models. A similar process is repeated for the non-chaotic masking sequence.

► Script A.3 is a comparative emulation of the stochastic features of the chaotic sequence relative to the Matlab internal PRNG[124].

► Script A.4 details the effects of despreader implementation losses when using reduced precision despreader arithmetic. Various methods of implementing the reduced-precision arithmetic are calculated and compared.

► Script A.5 analyzes receiver implementation losses, localized to the despreading process, as a result of timing and phase jitters.

► Script A.6 outlines the basic chaotic symbol normalization process that is applicable to constant-energy chaotic symbol modulation, intentionally varied symbol durations, and environmentally adaptive spreading ratios.

► Script A.7 is an exemplary adaptive correlator state machine.

A.1 Rayleigh Magnitude Nonlinear Processor Initialization

```
% 16-bit uniformly random input
x = 2^-17:2^-16:1;
% Comparative floating-point calculation
yideal = sqrt(-2*log(x));
% Fit actual curves
polydat = zeros(2,1024);
for ind = 1:64
    ind1 = 1025 + (ind-1)*32;
    ind2 = 1024 + ind*32;
    polydat(:,ind) = polyfit(x(ind1:ind2),yideal(ind1:ind2),1);
end
for ind = 65:1024
    ind1 = 3073 + (ind-65)*64;
    ind2 = 3072 + (ind-64)*64;
    polydat(:,ind) = polyfit(x(ind1:ind2),yideal(ind1:ind2),1);
end
%Truncate the coefficients to 18 bits
for ind = 1:15
    polydat(1,ind) = polydat(1,ind) + 16;
end
c1max = max(abs(polydat(1,:)));
c0max = max(abs(polydat(2,:)));
c1p2 = 2^(ceil(log2(c1max)));
c0p2 = 2^(ceil(log2(c0max)));
qpolydat = zeros(2,1024);
% Define output tables
adata = round((2^18/c1p2)*abs(polydat(1,:)))/(2^18/c1p2);
bdata = round((2^18/c0p2)*polydat(2,:))/(2^18/c0p2);
% Define output tables for endpoints
LUT_upper = 2^-15*round(2^15*sqrt(-2*log(64513*2^-16:2^-16:1)));
LUT_lower = 2^-15*round(2^15*sqrt(-2*log(2^-16:2^-16:2^-6)));
```

A.2 Digital Chaotic Circuit Initialization

```
% Specify the possible moduli values
p = [
    257 269 281 293 311 317 347 443 461 467 479 491 503 ...
    569 587 599 617 641 647 677 719 797 827 839 887 929 ...
    971 977 983 1013 1019 ...
    ];

% Specify the corresponding C values for the polynomial
g257=[110  118 0 0 0 0 0 0 0 0 ];      g269=[69  80 0 0 0 0 0 0 0 0 ];
g281=[95  248 0 0 0 0 0 0 0 0 ];       g293=[37  223 0 0 0 0 0 0 0 0 ];
g311=[107  169 0 0 0 0 0 0 0 0 ];      g317=[15  55 0 0 0 0 0 0 0 0 ];
g347=[89  219 0 0 0 0 0 0 0 0 ];       g443=[135  247  294  406 0 0 0 0 0 0];
g461=[240  323 0 0 0 0 0 0 0 0 ];      g467=[15  244  301  425 0 0 0 0 0 0];
g479=[233  352 0 0 0 0 0 0 0 0 ];      g491=[202  234 0 0 0 0 0 0 0 0 ];
g503=[8  271 0 0 0 0 0 0 0 0 ];        g569=[60  66 0 0 0 0 0 0 0 0 ];
g587=[245  472 0 0 0 0 0 0 0 0 ];      g599=[112  142  390  420 0 0 0 0 0 0];
g617=[158  390 0 0 0 0 0 0 0 0 ];      g641=[24  118 0 0 0 0 0 0 0 0 ];
g647=[169  190  421  427  579  585 0 0 0 0];
g677=[221  606 0 0 0 0 0 0 0 0 ];      g719=[488  630 0 0 0 0 0 0 0 0 ];
g797=[289  419 0 0 0 0 0 0 0 0 ];      g827=[83  376  596  690 0 0 0 0 0 0];
g839=[59  127 0 0 0 0 0 0 0 0 ];       g887=[383  405 0 0 0 0 0 0 0 0 ];
g929=[214  502  633  921 0 0 0 0 0 0]; g971=[675  835 0 0 0 0 0 0 0 0 ];
g977=[335  533 0 0 0 0 0 0 0 0 ];      g983=[78  140  311  554  647  890 0 0 0 0];
g1013=[77  823 0 0 0 0 0 0 0 0 ];      g1019=[272  592  653  973 0 0 0 0 0 0];

G = [ g257; g269; g281; g293; g311; g317; g347; g443; g461; g467; g479; ...
      g491; g503; g569; g587; g599; g617; g641; g647; g677; g719; g797; ...
      g827; g839; g887; g929; g971; g977; g983; g1013; g1019 ];

% Select 16 primes at random to serve as candidate primes -- may be
% considered as a session key.  More importantly, there exist Choose(31,16)
% possible combinations of primes, making selection 1 of 300,540,195.
q_temp1 = randperm(31);
q_temp2 = find( q_temp1 <= 16 );
p = p(q_temp2);

% Select 16 sufficiently random C* generating values
q_temp3a = floor( rand(1,16) * 1234567 );

for( qk = 1:16 )
    q_temp3b(qk) = length( find( G(q_temp2(qk),:) ~= 0 ) );
    C(qk) = G( q_temp2(qk) , mod( q_temp3a(qk) , q_temp3b(qk) ) + 1 );
end

% Select 16 sufficiently random initial conditions for the state jam vector
q_temp4 = floor( rand(1,16) * 1111111 );
state_jam_vec = mod( q_temp4 , p );

% Calculate the M for these moduli -- analysis only
capm = prod(p);

% Define the number of bits for desired Chaos generator output (max 18)
chaos_bits = 16;
```

```
% Initialize the max length LUTs for chaotic polynomial calculations
% cpv = chaotic_poly_values
cpv = zeros(16,1024);

% Calculate the table entries for the chaotic polynomial calculations
for( qk1 = 1:16 )
    cpv(qk1,:) = mod( mod( 3*[0:1023].^3 , p(qk1) ) + ...
                      mod( 3*[0:1023].^2 , p(qk1) ) + ...
                      [0:1023] + C(qk1) , p(qk1) );
%{
Eliminate the loop at the expense of slightly more computations (vectorized)
    for( qk2 = 0:(p(qk1)-1) )
        cpv(qk1,qk2+1) = mod( mod( 3*qk2.^3 , p(qk1) ) + ...
                              mod( 3*qk2.^2 , p(qk1) ) + ...
                              qk2 + C(qk1) , p(qk1) );
    end
%}
end

% Calculate the multiplication factors for CRT combination
GF2cb_mult_factors = zeros(1,16);
cb = chaos_bits;
for( qk1 = 1:12 )
  temp_p = p( find( [1:12] ~= qk1 ) );
  GF2cb_mult_factors(qk1) = ...
     mod( mod( mod( temp_p(1) * temp_p(2) * temp_p(3) , 2^cb ) * ...
            mod( temp_p(4) * temp_p(5) * temp_p(6) , 2^cb ) , 2^cb ) * ...
       mod( mod( temp_p(7) * temp_p(8) * temp_p(9) , 2^cb ) * ...
            mod( temp_p(10) * temp_p(11)           , 2^cb ) , 2^cb ) , 2^cb );
end

for( qk1 = 13:16 )
    temp_p = p( find( [13:16] ~= qk1 ) );
    GF2cb_mult_factors(qk1) = mod( prod(temp_p) , 2^cb );
end

%{
GF2cb_mult_factors(17) = 1;

% Puncture ring generator #17 to 2^8
puncture_temp = zeros(1,256);
cpv_temp = [];
for( qk2 = 1:p(17) )
    if( puncture_temp( mod(cpv(17,qk2),2^8)+1 ) == 0 )
        puncture_temp( mod(cpv(17,qk2),2^8)+1 ) = 1;
        cpv_temp = [cpv_temp qk2];
    end
end
cpv(17,:) = mod([cpv(cpv_temp) cpv(cpv_temp) cpv(cpv_temp) cpv(cpv_temp)],2^8);

%}
%Calculate the coefficients for the mixed-radix conversion
GF2cb = zeros( 16 , 1024 );
for( qk1 = 1:16 )
    GF2cb(qk1,:) = mod( GF2cb_mult_factors(qk1) * cpv(qk1,:) , 2^cb );
```

```
end
%}

% Generate random (permutation) ordering of orthogonal primes
% Equivalent combined mapping of approx 2^141, but uses fewer LUTs in the
% implementation that the traditional chaos generator.  The orderings may
% be thought of as defined by an initial key/initial condition rather than
% a defined chaotic polynomial -- the collectively prime combination
% retains the same guarantees for expanded key size.
q = [ 73 79 103 139 163 181 199 229 241 ]; % These primes must occur in pairs
q73 = randperm(73)-1;    q79 = randperm(79)-1;    q103 = randperm(103)-1;
q139 = randperm(139)-1;  q163 = randperm(163)-1;  q181 = randperm(181)-1;
q199 = randperm(199)-1;  q229 = randperm(229)-1;  q241 = randperm(241)-1;
q271 = randperm(271)-1;  q283 = randperm(283)-1;  q313 = randperm(313)-1;
q331 = randperm(331)-1;  q349 = randperm(349)-1;  q373 = randperm(373)-1;
q409 = randperm(409)-1;  q433 = randperm(433)-1;  q439 = randperm(439)-1;
q_sym = [q 512-fliplr(q)];
% Since the permutations define a unique ordering, there is no benefit to
% choosing an additional initial condition -- it is simply a compounded
% permutation operator that is no more unique than the original.
Q = [ q73 zeros(1,512-length(q73))    q79 zeros(1,512-length(q79))   ...
      q103 zeros(1,512-length(q103))  q139 zeros(1,512-length(q139))  ...
      q163 zeros(1,512-length(q163))  q181 zeros(1,512-length(q181)) ...
      q199 zeros(1,512-length(q199))  q229 zeros(1,512-length(q229)) ...
      q241 zeros(1,512-length(q241))  q271 zeros(1,512-length(q271)) ...
      q283 zeros(1,512-length(q283))  q313 zeros(1,512-length(q313)) ...
      q331 zeros(1,512-length(q331))  q349 zeros(1,512-length(q349)) ...
      q373 zeros(1,512-length(q373))  q409 zeros(1,512-length(q409)) ...
      q433 zeros(1,512-length(q433))  q439 zeros(1,512-length(q439))    ];
%Q = reshape(Q',18,512);

% Choose a random 8-bit controller and separate the remaining random primes
q_permute = q( floor( 9 * rand ) + 1 );
q_permute_complement = 512 - q_permute;
q_gen = setxor(q,q_permute);
q_complement = 512 - q_gen;

% Define location within Q to extract perm'd values
q_loc = zeros(1,18);
for( qk1 = 1:8 )
    q_loc(qk1) = find( q == q_gen(qk1) );
    q_loc(qk1+8) = 19-q_loc(qk1);
end
q_loc(17) = find( q == q_permute );
q_loc(18) = 19-q_loc(17);
```

A.3 Higher Order Statistics Emulation

```
%%% Initialization parameters
clear; close all; clc;
N = 1000000;     % Number of samples in sequence
MX = [];         % Retaining vector for quadrature chaos cumulants
MY = [];         % Retaining vector for Matlab Gaussian cumulants

for( k = 1:1000 )

%%% Digital chaotic sequence construction
% Generate uniform random numbers
A = (2^-16)*floor( (2^16)*rand(1,N) );
A( find(A == 0) ) = (2^-17)*ones( size( find(A == 0) ) );
B = (2^-16)*floor( (2^16)*rand(1,N) );

% Finite-precision Box-Muller transformation
M = sqrt( -2 * log(A) ) .* (1 + sign(randn(1,N)).*rand(1,N)/(2^14.5));
P = exp(j * 2 * pi * B) .* (1 + sign(randn(1,N)).*rand(1,N)/(2^15.5));

% Construct quadrature streams
XI = M .* real(P);
XQ = M .* imag(P);

%%% Generate Matlab Gaussian random variables
Y = randn(1,N);

%%% Calculate cumulant statistics
% Quadrature digital chaotic signal
m1X = mean(XI+j*XQ);
m1X_ = abs( m1X )/sqrt(2);
m2X = sum( ((XI-real(m1X)).^2 + (XQ-imag(m1X)).^2 ) )/length(XI)/2;
m3X = sum( ((XI-real(m1X)).^3 + (XQ-imag(m1X)).^3 ) )/length(XI)/2/(m2X^(3/2));
m4X = sum( ((XI-real(m1X)).^4 + (XQ-imag(m1X)).^4 ) )/length(XI)/2/(m2X^(4/2));
m5X = sum( ((XI-real(m1X)).^5 + (XQ-imag(m1X)).^5 ) )/length(XI)/2/(m2X^(5/2));
m6X = sum( ((XI-real(m1X)).^6 + (XQ-imag(m1X)).^6 ) )/length(XI)/2/(m2X^(6/2));
m7X = sum( ((XI-real(m1X)).^7 + (XQ-imag(m1X)).^7 ) )/length(XI)/2/(m2X^(7/2));
m8X = sum( ((XI-real(m1X)).^8 + (XQ-imag(m1X)).^8 ) )/length(XI)/2/(m2X^(8/2));

% Matlab generated Gaussian signal
m1Y = mean(Y);
m2Y = sum( (Y-m1Y).^2 )/length(Y);
m3Y = sum( (Y-m1Y).^3 )/length(Y)/(m2Y^(3/2));
m4Y = sum( (Y-m1Y).^4 )/length(Y)/(m2Y^(4/2));
m5Y = sum( (Y-m1Y).^5 )/length(Y)/(m2Y^(5/2));
m6Y = sum( (Y-m1Y).^6 )/length(Y)/(m2Y^(6/2));
m7Y = sum( (Y-m1Y).^7 )/length(Y)/(m2Y^(7/2));
m8Y = sum( (Y-m1Y).^8 )/length(Y)/(m2Y^(8/2));

% Store intermediate results
MX = [MX; m1X_ m2X m3X m4X m5X m6X m7X m8X];
MY = [MY; m1Y m2Y m3Y m4Y m5Y m6Y m7Y m8Y];

end
```

```
% Calculate absolute standard deviations
MXa = [ std(MX(:,1)) std(MX(:,2)) std(MX(:,3)) std(MX(:,4))
        std(MX(:,5)) std(MX(:,6)) std(MX(:,7)) std(MX(:,8)) ];
MYa = [ std(MY(:,1)) std(MY(:,2)) std(MY(:,3)) std(MY(:,4)) ...
        std(MY(:,5)) std(MY(:,6)) std(MY(:,7)) std(MY(:,8)) ];

% Calculate fractional standard deviations
MXf = [ std(MX(:,1)) std(MX(:,2)) std(MX(:,3)) std(MX(:,4))/3 ...
        std(MX(:,5))/sqrt(45) std(MX(:,6))/15 ...
        std(MX(:,7))/sqrt(1575) std(MX(:,8))/105 ];
MYf = [ std(MY(:,1)) std(MY(:,2)) std(MY(:,3)) std(MY(:,4))/3 ...
        std(MY(:,5))/sqrt(45) std(MY(:,6))/15 ...
        std(MY(:,7))/sqrt(1575) std(MY(:,8))/105 ];

save cumulant_compare MX MY MXa MYa MXf MYf

figure, plot( MXf , 'b-o' ), hold on, plot( MYf , 'g-x' )
```

A.4 Despreader Implementation Losses

```
%%% Evaluation of despreader implementation losses for chaotic waveforms
%%% resulting from reduced precision of the internally generated chaotic
%%% sequence.

%% Outline: Generation of time- and phase-synchronized chaotic signals

x = sqrt( -2 * log( rand(1,1000000) ) ) .* exp( j*2*pi* rand(1,1000000) );
y = conj(x);                          % Conjugate can go on either path WLOG

sym_energy_ideal = sum( x .* y );     % Expected value ~ 2000000

x_ones_c = [sign( real(x) ); abs(real(x)); sign(imag(x)); abs(imag(x))];

%% Scenario 1: Binary despreading sequence

x_binary = x_ones_c(1,:) + j*x_ones_c(3,:);
binary_energy = real( sum( y .* x_binary ) );

%% Scenario 2: Truncating to R MSBs

trunc_energy = zeros(1,16);           % Indices represent R MSBs retained
for( R = 1:16 )
    x_trunc_R = x_ones_c(1,:) .* floor( x_ones_c(2,:)*(2^(R-3)) ) / (2^(R-3)) + ...
             j*(x_ones_c(3,:) .* floor( x_ones_c(4,:)*(2^(R-3)) ) / (2^(R-3)) ) ;
    trunc_energy(R) = real( sum( y .* x_trunc_R ) );
end

%figure, plot( -10*log10( trunc_energy/sym_energy_ideal ) )

%% Scenario 3: Rounding to R MSBs

round_energy = zeros(1,16);          % Indices represent R MSBs retained
for( R = 1:16 )
    x_round_R = x_ones_c(1,:) .* round( x_ones_c(2,:)*(2^(R-3)) ) / (2^(R-3)) + ...
             j*(x_ones_c(3,:) .* round( x_ones_c(4,:)*(2^(R-3)) ) / (2^(R-3)) ) ;
    round_energy(R) = real( sum( y .* x_round_R ) );
end

%figure, plot( -10*log10( round_energy/sym_energy_ideal ) )

%% Scenario 4: Scaling R MSBs

scale_energy = zeros(1,16);
for( R = 1:16 )
    temp_r = scaled_value( x_ones_c(2,:) , R );
    temp_i = scaled_value( x_ones_c(4,:) , R );
    x_scale_R = x_ones_c(1,:) .* temp_r + ...
             j*(x_ones_c(3,:) .* temp_i );
    scale_energy(R) = real( sum( y .* x_scale_R ) );
end

%figure, plot( -10*log10( scale_energy/sym_energy_ideal ) )
```

```
%% Combined display of results

figure, hold on,
plot( 1:16 , -10*log10(ones(1,16) * binary_energy / sym_energy_ideal ), 'b-' ),
plot( 1:16 , -10*log10(trunc_energy / sym_energy_ideal ), 'r-o' ),
plot( 1:16 , -10*log10(round_energy / sym_energy_ideal ), 'g-x' ),
plot( 1:16 , -10*log10(scale_energy / sym_energy_ideal ), 'c-*' ),
legend('Binary chaotic despreading sequence',...
       'R truncated MSBs','R rounded MSBs',...
       'Magnitude scaled R MSBs')
ylabel('Receiver implementation loss'), xlabel('R MSBs retained')


%% Scaled value function
function [output] = scaled_value(input,scale)

output = zeros(size(input));
int_scale = 0;
for( k = 1:length(input) )
    if( ( input(k)/4 ) > 1 )
        int_scale = 1;
    elseif( ( input(k)/2 ) > 1 )
        int_scale = 2;
    elseif( ( input(k)/1 ) > 1 )
        int_scale = 3;
    elseif( ( input(k)*2 ) > 1 )
        int_scale = 4;
    elseif( ( input(k)*4 ) > 1 )
        int_scale = 5;
    elseif( ( input(k)*8 ) > 1 )
        int_scale = 6;
    elseif( ( input(k)*16 ) > 1 )
        int_scale = 7;
    elseif( ( input(k)*32 ) > 1 )
        int_scale = 8;
    elseif( ( input(k)*64 ) > 1 )
        int_scale = 9;
    elseif( ( input(k)*128 ) > 1 )
        int_scale = 10;
    elseif( ( input(k)*256 ) > 1 )
        int_scale = 11;
    elseif( ( input(k)*512 ) > 1 )
        int_scale = 12;
    elseif( ( input(k)*1024 ) > 1 )
        int_scale = 13;
    else
        int_scale = 13;
    end
    output(k) = round( input(k) * 2^(int_scale+scale-4) ) / (2^( int_scale+scale-4 ) );
end
```

A.5 Timing and Phase Jitter Susceptibility

```
%%% Timing and phase jitter susceptibility estimations

% Assumptions:
%  1. Timing and phase errors are Gaussian distributed random values
%  2. Phase jitter that causes symbol errors is self correcting since the
%  second-order loop filter operating on the phase errors is
%  unconditionally stable
%  3. Timing jitter that causes loss of lock (1 full chip outside either
%  the early or late detectors of the early-late gates) results in a
%  catastrophic loss of the signal that requires re-acquisition (this may
%  be mitigated by smart design).

x = sqrt( -2 * log( rand(1,1000000) ) ) .* exp( j*2*pi* rand(1,1000000) );
y = conj(x);                          % Conjugate can go on either path WLOG

sym_energy_ideal = sum( x .* y );    % Expected value ~ 2000000

%% Phase jitter estimates

jitters_rms = [0.1:0.1:10]/360;
phase_energy = zeros(size(jitters_rms));

for( k = 1:length( jitters_rms ) )
   phase_errors = exp( j*2*pi* jitters_rms(k) * randn( 1,1000000 ) );
   phase_energy(k) = sum( y .* x .* phase_errors );
end


%% Timing jitter estimates

jitters_time = [0.01:0.01:0.5];
time_energy = zeros(size(jitters_time));

for( k = 1:length( jitters_time ) )
    time_errors = sinc( jitters_time(k) * randn(1,1000000) );
    time_energy(k) = sum( y .* x .* time_errors );
end
```

A.6 Chaotic Symbol Normalization

```
%%% Evaluation of chaotic symbol normalization, equal energy transmission,
%%% and chaotic symbol dithering

%%% Equal energy chaotic symbols

Nsym = 10000;
Nchip = 800 * Nsym;
Threshold = 800;

Nsym_x = 0; Nsym_y = 0; Nsym_x_index = [1];
Esym_x = []; Esym_y = [];
E_count = 0;

x = (-2*log(rand(1,Nchip)));

for( k = 1:Nchip )
    if( mod(k,800) == 0 )
        Esym_y = [Esym_y sum(x( (k-799):k ) )];
    end
    E_count = E_count + x(k);
    if( E_count >= 2*Threshold )
        Nsym_x_index = [Nsym_x_index k];
        E_count = 0;
        Esym_x = [Esym_x sum(x( Nsym_x_index(end-1):Nsym_x_index(end) ) )];
    else
    end
end
```

A.7 Adaptive Correlator State Machine

```
function [wadr, we, int_radr, rcv_radr, freq_sel, freq_clr, mac_clr, mac_cap, ...
          mac_scale, phase_trig, acq_lock, acq_fail, delay_trig, state_tp]  = ...
          adapt_corr_ctrl(acq_strt, crse_thresh, med_thresh, fine_thresh, ...
          max_mag_sqrd, max_index, cent_grav)

%% Constants
NUM_CHAOS_SAMPS = 2048;
NUM_PRE_BUF_DELAY_SAMPS = 400;
NUM_POST_BUF_DELAY_SAMPS = 22;
NUM_CRSE_CORR_SAMPS = 32;
NUM_MED_CORR_SAMPS = 256 - 32;
NUM_FINE_CORR_SAMPS = 1024 - 256;
NUM_FINAL_CORR_SAMPS = 1536;

CORR_STEP_SIZE = 8;
CORR_CENTER = ceil(CORR_STEP_SIZE/2) - 1;
FINAL_CORR_STEP_SIZE = 1536/4;
MAX_CORR_OFFSET = 256;
SHRT_FREQ_SRCH = 1;

CRSE_CORR_SCALE = 5;   % ceil(log2(NUM_CRSE_CORR_SAMPS))
MED_CORR_SCALE = 8;    % ceil(log2(NUM_MED_CORR_SAMPS))
FINE_CORR_SCALE = 10;  % ceil(log2(NUM_FINE_CORR_SAMPS))
FINAL_CORR_SCALE = 11; % ceil(log2(NUM_FINAL_CORR_SAMPS))

FREQ_STRTUP_LATENCY = 3;
MAC_STRTUP_LATENCY = 8;
CORR_LATENCY = 9;
PEAK_DET_LATENCY = 25;

%% Persistent State Variable, Initialization, and Enumeration
persistent state;

if(isempty(state))  state = 0;    end

IDLE = 0;
PRE_BUF_DELAY = 1;
BUF_CHAOS = 2;
POST_BUF_DELAY = 3;
RUN_CRSE_CORR = 4;
CHK_CRSE_THRESH = 5;
UPDATE_TIME_FREQ = 6;
RUN_MED_CORR = 7;
CHK_MED_THRESH = 8;
RUN_FINE_CORR = 9;
CHK_FINE_THRESH = 10;
RUN_FINAL_CORR = 11;
DECLARE_LOCK = 12;

%% Persistent Variables and Initialization
persistent acq_fail_int;    persistent acq_lock_int;     persistent del_cnt;
```

```
persistent delay_trig_int;  persistent freq_clr_int;     persistent freq_sel_int;
persistent int_rd_cnt;      persistent mac_cap_int;      persistent mac_clr_int;
persistent mac_scale_int;   persistent phase_trig_int;   persistent rcv_rd_cnt;
persistent rd_cnt;          persistent we_int;           persistent wr_cnt;

if(isempty(acq_fail_int))    acq_fail_int = 0;    end
if(isempty(acq_lock_int))    acq_lock_int = 0;    end
if(isempty(del_cnt))         del_cnt = 0;         end
if(isempty(delay_trig_int))  delay_trig_int = 0;  end
if(isempty(freq_clr_int))    freq_clr_int = 0;    end
if(isempty(freq_sel_int))    freq_sel_int = 0;    end
if(isempty(int_rd_cnt))      int_rd_cnt = 0;      end
if(isempty(mac_cap_int))     mac_cap_int = 0;     end
if(isempty(mac_clr_int))     mac_clr_int = 0;     end
if(isempty(mac_scale_int))   mac_scale_int = 0;   end
if(isempty(phase_trig_int))  phase_trig_int = 0;  end
if(isempty(rcv_rd_cnt))      rcv_rd_cnt = 0;      end
if(isempty(rd_cnt))          rd_cnt = 0;          end
if(isempty(we_int))          we_int = 0;          end
if(isempty(wr_cnt))          wr_cnt = 0;          end

%% Quantization Settings
q_del_cnt = quantizer([9 0], 'wrap', 'ufixed');
q_delay_trig = quantizer([14 5], 'saturate', 'ufixed');
q_freq_sel = quantizer([3 0], 'wrap', 'ufixed');
q_mac_scale = quantizer([4 0], 'wrap', 'ufixed');
q_rd_cnt = quantizer([11 0], 'wrap', 'ufixed');
q_state = quantizer([4 0], 'wrap', 'ufixed');
q_wr_cnt = quantizer([11 0], 'wrap', 'ufixed');

%% State Machine
switch(state)
    case IDLE
        if(acq_strt == 1)
            acq_fail_int = 0;       acq_lock_int = 0;    del_cnt = 0;
            delay_trig_int = 0;     freq_sel_int = 4;    freq_clr_int = 0;
            int_rd_cnt = 0;         mac_cap_int = 0;     mac_clr_int = 0;
            mac_scale_int = 0;      phase_trig_int = 0;  rcv_rd_cnt = 0;
            rd_cnt = 0;             we_int = 0;          wr_cnt = 0;
            state = PRE_BUF_DELAY;
        else
            state = IDLE;
        end
    case PRE_BUF_DELAY
        if(del_cnt < NUM_PRE_BUF_DELAY_SAMPS - 1)
            del_cnt = quantize(q_del_cnt, del_cnt + 1);  we_int = 0;
            state = PRE_BUF_DELAY;
        else
            del_cnt = 0;    we_int = 1;
            state = BUF_CHAOS;
        end
    case BUF_CHAOS
        if(wr_cnt < NUM_CHAOS_SAMPS - 1)
            we_int = 1;    wr_cnt = quantize(q_wr_cnt, wr_cnt + 1);
            state = BUF_CHAOS;
        else
```

```
        we_int = 0;     wr_cnt = 0;
        state = POST_BUF_DELAY;
    end
case POST_BUF_DELAY
    if(del_cnt < NUM_POST_BUF_DELAY_SAMPS - 1)
        del_cnt = quantize(q_del_cnt, del_cnt + 1);
        state = POST_BUF_DELAY;
    else
        del_cnt = 0;
        state = RUN_CRSE_CORR;
    end
case RUN_CRSE_CORR
    if(rd_cnt < NUM_CRSE_CORR_SAMPS - 1)
        rd_cnt = quantize(q_rd_cnt, rd_cnt + 1);
        int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt + 1);
        rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt + 1);
        mac_scale_int = CRSE_CORR_SCALE;
        if(rd_cnt == FREQ_STRTUP_LATENCY - 1)
            freq_clr_int = 1;
        else
            freq_clr_int = 0;
        end
        if(rd_cnt == MAC_STRTUP_LATENCY - 1)
            mac_clr_int = 1;
        else
            mac_clr_int = 0;
        end
        state = RUN_CRSE_CORR;
    else
        if(del_cnt < CORR_LATENCY - 1)
            del_cnt = quantize(q_del_cnt, del_cnt + 1);
            state = RUN_CRSE_CORR;
        else
            del_cnt = 0;    mac_cap_int = 1;    rd_cnt = 0;
            state = CHK_CRSE_THRESH;
        end
    end
case CHK_CRSE_THRESH
    if(del_cnt < PEAK_DET_LATENCY - 1)
        del_cnt = quantize(q_del_cnt, del_cnt + 1);
        state = CHK_CRSE_THRESH;
    else
        del_cnt = 0;   mac_cap_int = 0;
        if(max_mag_sqrd >= crse_thresh)
            int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt + 1);
            rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt + 1);
            state = RUN_MED_CORR;
        else
            int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt-(NUM_CRSE_CORR_SAMPS-1));
            rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt-(NUM_CRSE_CORR_SAMPS-1));
            state = UPDATE_TIME_FREQ;
        end
    end
case UPDATE_TIME_FREQ
    if((rcv_rd_cnt - int_rd_cnt) >= MAX_CORR_OFFSET)
        acq_fail_int = 1;
```

```
        state = IDLE;
    else
        if(freq_sel_int == 0)
            freq_sel_int = 4;
            rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt+CORR_STEP_SIZE);
        else
            freq_sel_int = quantize(q_freq_sel,freq_sel_int-1-SHRT_FREQ_SRCH);
        end
        state = RUN_CRSE_CORR;
    end
case RUN_MED_CORR
    if(rd_cnt < NUM_MED_CORR_SAMPS - 1)
        rd_cnt = quantize(q_rd_cnt, rd_cnt + 1);
        int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt + 1);
        rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt + 1);
        mac_scale_int = MED_CORR_SCALE;
        if(rd_cnt == FREQ_STRTUP_LATENCY - 1)
            freq_clr_int = 1;
        else
            freq_clr_int = 0;
        end
        if(rd_cnt == MAC_STRTUP_LATENCY - 1)
            mac_clr_int = 1;
        else
            mac_clr_int = 0;
        end
        state = RUN_MED_CORR;
    else
        if(del_cnt < CORR_LATENCY - 1)
            del_cnt = quantize(q_del_cnt, del_cnt + 1);
            state = RUN_MED_CORR;
        else
            del_cnt = 0;     mac_cap_int = 1;       rd_cnt = 0;
            state = CHK_MED_THRESH;
        end
    end
case CHK_MED_THRESH
    if(del_cnt < PEAK_DET_LATENCY - 1)
        del_cnt = quantize(q_del_cnt, del_cnt + 1);
        state = CHK_MED_THRESH;
    else
        del_cnt = 0;    mac_cap_int = 0;
        if(max_mag_sqrd >= med_thresh)
            int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt + 1);
            rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt + 1);
            state = RUN_FINE_CORR;
        else
            int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt ...
                - (NUM_CRSE_CORR_SAMPS - 1) - NUM_MED_CORR_SAMPS);
            rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt ...
                - (NUM_CRSE_CORR_SAMPS - 1) - NUM_MED_CORR_SAMPS);
            state = UPDATE_TIME_FREQ;
        end
    end
case RUN_FINE_CORR
    if(rd_cnt < NUM_FINE_CORR_SAMPS - 1)
```

```
        rd_cnt = quantize(q_rd_cnt, rd_cnt + 1);
        int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt + 1);
        rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt + 1);
        mac_scale_int = FINE_CORR_SCALE;
        if(rd_cnt == FREQ_STRTUP_LATENCY - 1)
            freq_clr_int = 1;
        else
            freq_clr_int = 0;
        end
        if(rd_cnt == MAC_STRTUP_LATENCY - 1)
            mac_clr_int = 1;
        else
            mac_clr_int = 0;
        end
        state = RUN_FINE_CORR;
    else
        if(del_cnt < CORR_LATENCY - 1)
            del_cnt = quantize(q_del_cnt, del_cnt + 1);
            state = RUN_FINE_CORR;
        else
            del_cnt = 0;   mac_cap_int = 1;   rd_cnt = 0;
            state = CHK_FINE_THRESH;
        end
    end
case CHK_FINE_THRESH
    if(del_cnt < PEAK_DET_LATENCY - 1)
        del_cnt = quantize(q_del_cnt, del_cnt + 1);
        state = CHK_FINE_THRESH;
    else
        del_cnt = 0;    mac_cap_int = 0;
        if(max_mag_sqrd >= fine_thresh)
            delay_trig_int = quantize(q_delay_trig, rcv_rd_cnt - ...
                (NUM_CRSE_CORR_SAMPS - 1) - NUM_MED_CORR_SAMPS - ...
                NUM_FINE_CORR_SAMPS + max_index);
            int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt - (NUM_CRSE_CORR_SAMPS - 1) ...
                - NUM_MED_CORR_SAMPS - NUM_FINE_CORR_SAMPS + CORR_CENTER);
            rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt - (NUM_CRSE_CORR_SAMPS - 1) ...
                - NUM_MED_CORR_SAMPS - NUM_FINE_CORR_SAMPS + max_index);
            state = RUN_FINAL_CORR;
        else
            int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt - (NUM_CRSE_CORR_SAMPS - 1) ...
                - NUM_MED_CORR_SAMPS - NUM_FINE_CORR_SAMPS);
            rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt - (NUM_CRSE_CORR_SAMPS - 1) ...
                - NUM_MED_CORR_SAMPS - NUM_FINE_CORR_SAMPS);
            state = UPDATE_TIME_FREQ;
        end
    end
case RUN_FINAL_CORR
    if(rd_cnt < NUM_FINAL_CORR_SAMPS - 1)
        rd_cnt = quantize(q_rd_cnt, rd_cnt + 1);
        int_rd_cnt = quantize(q_rd_cnt, int_rd_cnt + 1);
        rcv_rd_cnt = quantize(q_rd_cnt, rcv_rd_cnt + 1);
        mac_scale_int = FINAL_CORR_SCALE;
        if(rd_cnt == FREQ_STRTUP_LATENCY - 1)
            freq_clr_int = 1;
        else
```

```
            freq_clr_int = 0;
        end
        if(rd_cnt == MAC_STRTUP_LATENCY - 1)
            mac_clr_int = 1;
        else
            mac_clr_int = 0;
        end
        if((rd_cnt == (FINAL_CORR_STEP_SIZE - 1) + (CORR_LATENCY - 1)) || ...
           (rd_cnt == ((2 * FINAL_CORR_STEP_SIZE) - 1) + (CORR_LATENCY - 1)) || ...
           (rd_cnt == ((3 * FINAL_CORR_STEP_SIZE) - 1) + (CORR_LATENCY - 1)))
            phase_trig_int = 1;
        else
            phase_trig_int = 0;
        end
        state = RUN_FINAL_CORR;
    else
        if(del_cnt < CORR_LATENCY - 1)
            if(del_cnt == CORR_LATENCY - 2)
                phase_trig_int = 1;
            end
            del_cnt = quantize(q_del_cnt, del_cnt + 1);
            state = RUN_FINAL_CORR;
        else
            del_cnt = 0;  mac_cap_int = 1;  phase_trig_int = 0;  rd_cnt = 0;
            state = DECLARE_LOCK;
        end
    end
case DECLARE_LOCK
    if(del_cnt < PEAK_DET_LATENCY - 1)
        del_cnt = quantize(q_del_cnt, del_cnt + 1);
        state = DECLARE_LOCK;
    else
        acq_lock_int = 1;   del_cnt = 0;
        delay_trig_int = quantize(q_delay_trig, delay_trig_int + cent_grav - 3);
        state = IDLE;
    end
otherwise
    state = IDLE;
end

%% Quantized Outputs
acq_fail = acq_fail_int;   acq_lock = acq_lock_int;   delay_trig = delay_trig_int;
freq_clr = freq_clr_int;   freq_sel = freq_sel_int;   int_radr = int_rd_cnt;
mac_cap = mac_cap_int;     mac_clr = mac_clr_int;     rcv_radr = rcv_rd_cnt;
mac_scale = quantize(q_mac_scale, mac_scale_int);
phase_trig = phase_trig_int;        state_tp = quantize(q_state, state);
wadr = wr_cnt;   we = we_int;
```

Appendix B: Chaotic Communications Patent Listing

The work presented in this document is the subject of over 30 pending U.S. patent applications[108, 109, 159, 93, 94, 95, 156, 96, 160, 157, 201, 97, 98, 99, 100, 158, 110, 115, 173, 174, 198, 199, 188, 189, 190, 192, 193, 194, 202, 169, 165, 197, 203, 195, 196]; as of publishing this dissertation, 8 of the applications listed in Table 10 and Table 11 were posted to the U.S.P.T.O. website.

Table 10. Chaotic communications (pending) patent listing (part I)

Harris ID	**Title**	Inventor(s)
GCSD-1854	Encryption via induced unweighted errors (20080294956)	Chester/Michaels
GCSD-1856	Digital generation of a chaotic numerical sequence (20080263119)	Chester/Michaels
GCSD-1874	Spread spectrum communications system and method utilizing chaotic sequence (20080304666)	Chester/Michaels
GCSD-1907	Extending a repetition period of a random sequence (20080294710)	Michaels/Chester
GCSD-1918	Mixed radix conversion with a priori defined statistical artifacts (20080307022)	Michaels/Chester
GCSD-1936	Closed Galois Field combination (20090044080)	Michaels/Chester
GCSD-1952	Adaptive correlation	Chester/Michaels
GCSD-1969	Mixed radix conversion with chosen statistical artifacts (20080307024)	Michaels/Chester
GCSD-1990	Chaotic spread spectrum communications system receiver (20090034727)	Chester/Michaels
GCSD-1993	Featureless coherent chaotic amplitude modulation	Chester/Michaels
GCSD-2017	Cryptographic system incorporating a digitally generated chaotic numerical sequence	Michaels
GCSD-2018	Cryptographic system configured for extending a repetition period of a random sequence	Michaels
GCSD-2019	Cryptographic system configured to perform a mixed-radix conversion with a priori defined statistical artifacts	Michaels/Chester
GCSD-2020	Cryptographic system including a mixed-radix number generator with chosen statistical artifacts	Michaels/Chester
GCSD-2021	A closed Galois Field cryptographic system	Michaels
GCSD-2022	Selective noise cancellation of a spread spectrum signal	Michaels
GCSD-2044	Improved efficiency pseudo-chaos generation	Michaels/Chester
GCSD-2045	Improved efficiency sine/cosine generation	Michaels/Chester
GCSD-2055	CAZAC-based anti-jam waveform	Michaels/Chester
GCSD-2056	Chaotic AJ waveform in HF ELOS applications	Michaels/Chester

TABLE 11. Chaotic communications (pending) patent listing (part II)

Harris ID	**Title**	Inventor(s)
GCSD-2058	Continuous time chaos dithering	Chester/Michaels
GCSD-2059	Discrete time chaos dithering	Chester/Michaels
GCSD-2060	Chaotic multiple access communications: shared sequences, coordinated sequence offsets	Michaels/Chester
GCSD-2061	Chaotic multiple access communications: independent orthogonal sequences	Michaels/Chester
GCSD-2062	Chaotic multiple access communications: coordinated orthogonal sequences	Michaels/Chester
GCSD-2063	Protected chaos-based multiple access communications: additive Gaussian masking	Michaels/Chester
GCSD-2064	Protected chaos-based multiple access communications: stationary and non-stationary phase rotations	Michaels/Chester
GCSD-2065	Protected chaos-based multiple access communications: time-division sequence modulation	Michaels/Chester
GCSD-2066	High-speed chaos-based cryptographic system	Michaels/Chester
GCSD-2067	RAKE receiver for chaotic communications	Michaels/Chester
GCSD-2068	Normalization techniques for non-stationary chaotic communications waveforms	Michaels/Chester
GCSD-2069	Symbol duration dithering for secured chaotic communications	Michaels/Chester
GCSD-2070	Constant energy symbols in chaotic communications for optimal BER performance	Michaels/Chester
GCSD-2071	Multi-tier adhoc networking neighbor discovery techniques	Michaels
GCSD-2072	Multi-tier DTDMA communications	Michaels

References

[1] C. Shannon, "Communication in the presence of noise," *Proc. Inst. Radio Eng.*, vol. 37, pp. 10–21, Jan 1947.

[2] L. Chua, "Dynamic nonlinear networks: State-of-the-art," *IEEE Transactions on Circuits and Systems*, vol. 27, pp. 1059–1087, Nov 1980.

[3] H. Dedieu, M. Kennedy, and M. Hasler, "Chaos shift keying: modulation and demodulation of a chaotic carrier using self-synchronizing chua's circuits," *IEEE Transactions on Circuits and Systems II*, vol. 40, pp. 634–642, 1993.

[4] R. Pickholtz, D. Schilling, and L. Milstein, "Theory of spread spectrum communications – a tutorial," *IEEE Transactions on Communications*, vol. 30, pp. 855–884, May 1982.

[5] G. Kolumban, M. Kennedy, and L. Chua, "The role of synchronization in digital communications using chaos: Fundamentals of digital communications," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 10, pp. 927–936, Oct 1997.

[6] M. Hasler and Y. Maistrenko, "An introduction to the synchronization of chaotic systems: coupled skew tent maps," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 10, pp. 856–866, Oct 1997.

[7] R. Scholtz, "The origins of spread-spectrum communications," *IEEE Transactions on Communications*, vol. 30, pp. 822–854, 1982.

[8] E. Appleton and M. Barnett, "On some direct evidence for downward atmospheric reflection of electric rays," *Proceedings Royal Society Series A*, vol. 109, pp. 621–641, Dec 1925.

[9] J. Boone and R. Peterson, "SIGSALY: the start of the digital revolution," *http://www.nsa.gov/publications/publi00019.cfm*, 23 Apr 2007.

[10] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[11] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, Oct 1949.

[12] R. Pickholtz, L. Milstein, and D. Schilling, "Spread spectrum for mobile communications," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 313–322, May 1991.

[13] G. Stüber, *Principles of Mobile Communication*. Kluwer, second ed., 2001.

[14] K. Cuomo and A. Oppenheim, "Circuit implementation of synchronized chaos with applications to communications," *Physics Review Letters*, vol. 71, pp. 65–68, 1993.

[15] K. Cuomo and A. Oppenheim, "Synchronization of Lorenz-based chaotic circuits with applications to communications," *IEEE Transactions on Circuits and Systems*, vol. 40, pp. 626–633, 1993.

[16] C. Wu and L. Chua, "A simple way to synchronize chaotic systems with applications to secure communication systems," *International Journal of Bifurcations and Chaos*, vol. 3, pp. 1619–1627, 1994.

[17] E. Weisstein, "Chaos," *http://mathworld.wolfram.com/Chaos.html*, 23 Apr 2007.

[18] R. Bartle, *Elements of Real Analysis*. Wiley, second ed., 1976.

[19] L. Debnath and P. Mikusinski, *Introduction to Hilbert Spaces with Applications*. Academic Press, second ed., 1998.

[20] R. Millikan, "A new modification of the cloud method of determining the elementary electrical charge and the most probable value of that charge," *The London, Edinburgh, and Dublin Philisophical Magazine*, vol. XIX, pp. 209–228, 1910.

[21] T. Yoshimura and T. Kohda, "Resonance properties of Chebyshev chaotic sequences," *ISCAS Proceedings*, vol. 4, pp. 573–576, 23-26 May 2004.

[22] D. Xiao, "Using Chebyshev chaotic map to construct infinite length hash chains," *IEEE ICCCAS*, vol. 1, pp. 11–12, 2004.

[23] P. Ashwin and I. Melbourne, "Symmetry groups of attractors," *Archive for Rational Mechanics and Analysis*, vol. 126, no. 1, pp. 59–78, 1994.

[24] P. Verhulst, "Reserches mathematiques sur la loi d'accroissement de la population," *Nouv. mem de l'Academie Royale des Sci et Belles-Lettres de Bruxelles*, vol. 18, pp. 1–41, 1845.

[25] M. Panella and G. Martinelli, "An RNS architecture for quasi-chaotic oscillators," *The Journal of VLSI Signal Processing*, vol. 33, no. 1-2, pp. 199–220, Jan 2003.

[26] T. Matsumoto, L. Chua, and S. Tanaka, "Simplest chaotic nonautonomous circuit," *Physical Review A*, vol. 30, pp. 1155–1157, 1984.

[27] T. Matsumoto, "A chaotic attractor from Chua's circuit," *IEEE Transactions on Circuits and Systems*, vol. 31, pp. 1055–1058, Dec 1984.

[28] A. Rodriguez-Vazquez and M. Delgado-Restituto, "CMOS design of chaotic oscillators using state variables: a monolithic Chua's circuit," *IEEE Transactions on Circuits and Systems-II: Analog and DSP*, vol. 40, no. 10, pp. 596–613, 1993.

[29] H. Song and K. Kwack, "CMOS circuit design and implementation of the discrete time chaotic chip," *IEEE Transactions on Circuits and Systems III*, pp. 73–76, 2002.
[30] A. Elwakil and M. Kennedy, "Improved implementation of Chua's chaotic oscillator using currentfeedback op amp," *IEEE Transaction on Circuits and Systems I*, vol. 47, no. 1, pp. 76–79, 2000.
[31] E. Ott, C. Grebogi, and J. Yorke, "Controlling chaos," *Physics Review Letters*, vol. 64, pp. 1196–1199, Mar 1990.
[32] L. Pecora and T. Carroll, "Synchronization in chaotic systems," *Physics Review Letters*, vol. 64, pp. 821–824, Feb 1990.
[33] T. Carroll and L. Pecora, "Synchronizing chaotic circuits," *IEEE Transactions on Circuits and Systems*, vol. 38, pp. 453–456, 1991.
[34] L. Pecora, T. Carroll, G. Johnson, D. Mar, and J. Heagy, "Fundamentals of synchronization in chaotic systems, concepts, and applications," *Chaos*, vol. 7, pp. 520–543, Dec. 1997.
[35] G. Freeland and T. Durrani, "Multipredictor modelling with application to chaotic signals," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 133–136 vol.3, Apr 1993.
[36] Y.-S. Lau, K. Lin, and Z. Hussain, "Space-time encoded secure chaos communications with transmit beamforming," *TENCON 2005 IEEE Region 10*, pp. 1–5, Nov. 2005.
[37] M. Ciftci and D. Williams, "Iterative equalization for chaotic communications systems," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 165–168, March 2005.
[38] G. Bateni and C. McGillem, "A chaotic direct-sequence spread-spectrum communications sequence," *IEEE Transactions on Communications*, vol. 42, pp. 1524–1527, 1994.
[39] T. Yang and L. Chua, "Secure communication via chaotic parameter modulation," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 43, no. 9, pp. 817–819, Sep 1996.
[40] F. Lau and C. Tse, *Chaos-Based Digital Communication Systems*. Springer, first ed., 2003.
[41] T. Yang, "A survey of chaotic secure communication systems," 2004.
[42] W. Tam, F. Lau, and C. Tse, *Digital Communications with Chaos: Multiple Access Techniques and Performance*. Elsevier, first ed., 2007.
[43] G. Kolumban, M. Kennedy, and L. Chua, "The role of synchronization in digital communications using chaos: Chaotic modulation and chaotic synchronization," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, no. 11, pp. 1129–1140, Nov 1998.
[44] G. Kolumban and M. Kennedy, "The role of synchronization in digital communications using chaos: Performance bounds for correlation receivers," *IEEE Transactions on Circuits and Systems I*, vol. 47, pp. 1673–1683, 2000.
[45] N. Rulkov, M. Sushchik, L. Tsimring, and H. Abarbanel, "Generalized synchronization of chaos in directionally coupled chaotic systems," *Physics Review Letters*, vol. 64, pp. 980–994, Feb 1995.
[46] M. Kennedy, "Three steps to chaos: a Chua's circuit primer," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 10, pp. 657–674, Oct 1993.
[47] J. Mankins, "Technical readiness levels," *Office of Space Access and Technology, NASA*, pp. 1–5, Apr 1995.
[48] L. Chua, M. Komouro, and T. Matsumoto, "The double scroll family," *IEEE Transactions on Circuits and Systems*, vol. 33, pp. 1072–1118, 1986.
[49] E. Lorenz, "Deterministic non-periodic flows," *Journal of Atmospheric Sciences*, vol. 20, pp. 130–141, 1963.
[50] O. Rössler, "An equation for continuous chaos," *Physics Letters*, vol. 57A, pp. 397–398, 1976.
[51] J. Sprott, "A new class of chaotic circuit," *Physics Letters A*, vol. 266, pp. 19–23, Jan 2000.
[52] K. Kiers, D. Schmidt, and J. Sprott, "Precision measurements of a simple chaotic circuit," *American Journal of Physics*, vol. 72, pp. 503–509, Apr 2004.
[53] L. Fortuna, M. Frasca, S. Graziani, and S. Reddiconto, "A chaotic circuit with ferroelectric nonlinearity," *Nonlinear Dynamics*, vol. 44, pp. 55–61, 2006.
[54] J. Martin-Pereda, A. Gonzalez-Marcos, and C. Sanchez-Guillen, "Synchronizing chaotic optically-programmable digital circuits," *Global Telecommunications Conference*, vol. 3, pp. 2078–2082, Nov 1996.
[55] R. Senani and S. Gupta, "Implementation of Chua's chaotic circuit using current feedback op-amps," *Electronics Letters*, vol. 34, pp. 829–830, Apr 1998.
[56] A. Elwakil and A. Soliman, "Current mode chaos generator," *Electronics Letters*, vol. 33, no. 20, pp. 1661–1662, 1997.
[57] Y. Uchitani and Y. Nishio, "Synchronization patterns generated in a ring of cross-coupled chaotic circuits," *IEEE International Joint Conference on Neural Networks*, pp. 3855–3860, June 2008.
[58] H. Sekiya, M. Noguchi, S. Moro, and S. Mori, "Synchronization in chaotic circuits mutually full-coupled by capacitors," *ISCAS Proceedings*, vol. 2, pp. 817–820, Jun 1997.
[59] E. Sanchez, M. Matias, and V. Perez-Munuzuri, "An experimental setup for studying the effect of noise on Chua's circuit," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, pp. 517–520, Apr 1999.

[60] O. Saracoglu and R. Kilic, "A simulation study on EMI effects in autonomous Chua's chaotic circuit," *IEEE International Symposium on Electromagnetic Compatibility*, vol. 1, pp. 280–283, May 2003.
[61] S. Gregori and A. Cabrini, "CMOS discrete-time chaotic circuit for low-power embedded cryptosystems," *48th Midwest Symposium on Circuits and Systems*, vol. 2, pp. 1498–1501, Aug. 2005.
[62] Z. Liu, S. Yu, G. Xie, and Y. Liu, "A novel fourth-order chaotic circuit and its implementation," *The 9th International Conference for Young Computer Scientists*, pp. 3045–3050, Nov. 2008.
[63] P. Zhou and Q. Huang, "A chaotic circuit and some results," *International Conference on Communications, Circuits and Systems*, vol. 4, pp. 2353–2355, June 2006.
[64] R. Rovatti, G. Setti, and G. Mazzini, "Chaotic complex spreading sequences for asynchronous DS-CDMA: Some theoretical performance bounds," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, pp. 496–506, Apr 1998.
[65] W. Tam, F. Lau, and C. Tse, "Exact analytical bit error rates for multiple access chaos-based communication systems," *IEEE Transactions on Circuits and Systems II*, vol. 51, pp. 473–481, Sep 2004.
[66] D. Sandoval-Morantes and D. Munoz-Rodriguez, "Chaotic sequences for multiple access," *Electronics Letters*, vol. 34, pp. 235–237, Feb 1998.
[67] S. Hayes, E. Grebogi, E. Ott, and A. Mark, "Experimental control of chaos for communication," *Physics Review Letters*, vol. 73, p. 1781, Sept. 1994.
[68] S. Hayes, S. Grebogi, and E. Ott, "Communication with chaos," *Physics Review Letters*, vol. 70, pp. 3032–3034, May 1993.
[69] G. Kolumban, M. Kennedy, Z. Jako, and G. Kis, "Chaotic communications with correlator receivers: Theory and performance limits," *Proceeding of the IEEE*, vol. 90, pp. 711–732, 2002.
[70] G. Bernstein and M. Lieberman, "Secure random number generation using chaotic circuits," *IEEE Transactions on Circuits and Systems*, vol. 37, pp. 1157–1164, Sep 1990.
[71] D. Leon, S. Balkir, M. Hoffman, and L. Perez, "Pseudo-chaotic PN-sequence generator circuits for spread spectrum communications," *IEEE Proceedings on Circuits, Devices and Systems*, vol. 151, pp. 543–550, Dec. 2004.
[72] Y. Andreyev and E. Efremova, "Separation of wideband chaotic signals," *International Symposium on Signals, Circuits and Systems*, vol. 1, pp. 1–4, July 2003.
[73] M. Panella and G. Martinelli, "RNS quasi-chaotic generator for self-correcting secure communication," *Electronics Letters*, vol. 37, pp. 325–327, Mar 2001.
[74] M. Panella and G. Martinelli, "RNS quasi-chaotic generators," *Electronics Letters*, vol. 36, pp. 1325–1326, Jul 2000.
[75] D. Petruzzi, J. Moreira, and D. Levin, "Quasi-chaotic coding over GF(q)," *IEEE Transactions on Communications*, vol. 54, pp. 462–468, March 2006.
[76] D. Frey, "Chaotic digital encoding: an approach to secure communication," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, pp. 660–666, Oct 1993.
[77] G. Cardarilli, M. Re, and R. Lojacono, "RNS-to-binary conversion for efficient VLSI implementation," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, pp. 667–669, Jun 1998.
[78] E. Di Claudio, G. Orlandi, and F. Piazza, "Fast RNS DSP algorithms implemented with binary arithmetic," *International Conference on Acoustics, Speech, and Signal Processing*, pp. 1531–1534 vol.3, Apr 1990.
[79] J. Cooley and J. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, pp. 297 – 301, 1965.
[80] R. Singleton, "An algorithm for computing the mixed radix fast Fourier transform," *IEEE Transactions on Audio and Electroacoustics*, vol. 17, no. 2, pp. 93 – 103, 1969.
[81] G. Cardarilli, A. Del Re, A. Nannarelli, and M. Re, "Programmable power-of-two RNS scaler and its application to a QRNS polyphase filter," *IEEE International Symposium on Circuits and Systems*, pp. 1102–1105 Vol. 2, May 2005.
[82] Q. Ke and M. Feldman, "Single flux quantum circuits using the residue number system," *IEEE Transactions on Applied Superconductivity*, vol. 5, pp. 2988–2991, Jun 1995.
[83] K. Rosen, *Elementary Number Theory and its Applications*. Addison Wesley Longman, fourth ed., 2000.
[84] T. Apostol, *Introduction to Analytic Number Theory*. Springer, first ed., 1998.
[85] N. Szabo and R. Tanaka, *Residue arithmetic and its application to computer technology*. McGraw-Hill, first ed., 1967.
[86] M. Soderstrand, W. Jenkins, G. Jullien, and F. Taylor, *Residue number system arithmetic: modern applications in digital signal processing*. IEEE press, reprint ed., 1986.
[87] E. Weisstein, "Rings," *http://mathworld.wolfram.com/Ring.html*, 12 Aug 2007.
[88] E. Weisstein, "Fields," *http://mathworld.wolfram.com/Field.html*, 12 Aug 2007.
[89] IEEE, "IEEE standard for binary floating-point arithmetic," *ANSI/IEEE Std 754-1985*, Aug 1985.

[90] E. Weisstein, "Prime factorization algorithms," *http://mathworld.wolfram.com/PrimeFactorizationAlgorithms.html*, 17 Mar 2008.
[91] A. Booth, "A signed binary multiplication technique," *Journal of Mechanics and Applied Mathematics*, pp. 236–240, Jun 1951.
[92] B. Parhami, "Generalized signed-digit number systems: a unifying framework for redundant number systems," *IEEE Transactions on Computers*, vol. 39, pp. 89–98, Jan 1990.
[93] A. Michaels and D. Chester, *Extending a Repetition Period of a Random Sequence*. Harris Corporation patent (pending) GCSD-1907, 2008.
[94] A. Michaels and D. Chester, *Mixed Radix Conversion with a priori Defined Statistical Artifacts*. Harris Corporation patent (pending) GCSD-1918, 2007.
[95] A. Michaels and D. Chester, *Closed Galois Field Combination*. Harris Corporation patent (pending) GCSD-1936, 2007.
[96] A. Michaels and D. Chester, *Mixed Radix Conversion with Chosen Statistical Artifacts*. Harris Corporation patent (pending) GCSD-1969, 2008.
[97] A. Michaels, *Cryptographic System Configured for Extending a Repetition Period of a Random Sequence*. Harris Corporation patent (pending) GCSD-2018, 2008.
[98] A. Michaels and D. Chester, *Cryptographic System Configured to Perform a Mixed-Radix Conversion with a priori Defined Statistical Artifacts*. Harris Corporation patent (pending) GCSD-2019, 2008.
[99] A. Michaels and D. Chester, *Cryptographic System including a Mixed Radix Number Generator with Chosen Statistical Artifacts*. Harris Corporation patent (pending) GCSD-2020, 2008.
[100] A. Michaels, *A Closed Galois Field Cryptographic System*. Harris Corporation patent (pending) GCSD-2021, 2008.
[101] K. Konishi, M. Ishii, and H. Kokame, "Stabilizing unstable periodic points of one-dimensional nonlinear systems," *Physical Review E*, vol. 54, pp. 3455 – 3460, 1996.
[102] R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *IEEE Transactions on Information Theory*, vol. 13, no. 4, pp. 619–621, Oct 1967.
[103] GPS Joint Program Office, "Navstar GPS Space Segment / Navigation User Interfaces, Rev D," pp. 20–29, Dec 2004.
[104] E. Zenner, "Cryptanalysis of LFSR-based pseudorandom generators: a survey," *Reihe Informatik*, p. 29, 2004.
[105] S. Li, C. Li, K. Lo, and G. Chen, "Cryptanalyzing and encryption scheme based on blind source separation," *IEEE Transaction on Circuits and Systems I*, vol. 55, pp. 1055–1063, May 2008.
[106] S. Halevi and C. Jutla, "Cryptanalysis of stream ciphers with linear masking," in *Proceedings of CRYPTO02*, pp. 515–532, Springer-Verlag, 2002.
[107] C. Paar, "A new architecture for a parallel finite field multiplier with low complexity based on composite fields," *IEEE Transactions on Computers*, vol. 45, no. 7, pp. 856–861, Jul 1996.
[108] D. Chester and A. Michaels, *Encryption Via Induced Unweighted Errors*. Harris Corporation patent (pending) GCSD-1854, 2007.
[109] D. Chester and A. Michaels, *Digital Generation of a Chaotic Numerical Sequence*. Harris Corporation patent (pending) GCSD-1856, 2007.
[110] A. Michaels and D. Chester, *Improved Efficiency Pseudo-Chaos Generation*. Harris Corporation patent (pending) GCSD-2044, 2008.
[111] G. Box and M. Muller, "A note on the generation of random normal deviates," *Princeton University*, pp. 610–611, Jan 1958.
[112] L. Devroye, *Non-uniform random variate generation*. Springer, first ed., 1986.
[113] F. Yongquan and Z. Zilic, "A novel scheme of implementing high speed AWGN communication channel emulators in FPGAs," *International Symposium on Circuits and Systems*, vol. 2, pp. II–877–80 Vol.2, 23-26 May 2004.
[114] J. Fox, W. Young, and D. Chester, *Sine/cosine generator and method*. U.S.P.T.O. patent 5276633, 1994.
[115] A. Michaels and D. Chester, *Improved Efficiency Sine/Cosine Generation*. Harris Corporation patent (pending) GCSD-2045, 2008.
[116] A. Katz, "Linearization: reducing distortion in power amplifiers," *IEEE Microwave Magazine*, vol. 2, pp. 37–49, Dec 2001.
[117] S. Han and J. Lee, "An overview of peak-to-average power ratio reduction techniques for multicarrier transmission," *IEEE Wireless Communications Magazine*, vol. 12, pp. 56–65, Apr 2005.
[118] V. Jain and L. Lin, "High-speed double precision computation of nonlinear functions," in *ARITH '95: Proceedings of the 12th Symposium on Computer Arithmetic*, p. 107, IEEE Computer Society, 1995.
[119] V. Jain, S. Shrivastava, A. Snider, D. Damerow, and D. Chester, "Hardware implementation of a nonlinear processor," in *IEEE International Symposium on Circuits and Systems*, vol. 6, pp. 509–514, 1999.
[120] Xilinx, "Virtex-4 family overview," pp. 1–9, 2007.

[121] D. Knuth, *Art of Computer Programming, Volume II: Seminumerical Algorithms*. Addison-Wesley Professional, third ed., 1997.
[122] G. Marsaglia, "DIEHARD battery of tests of randomness," 1995.
[123] R. V. Hogg and A. Craig, *Introduction to Mathematical Statistics*. Prentice Hall, fifth ed., 1994.
[124] M. Matsumoto, "Mersenne twister homepage," *http://www.math.sci.hiroshima-u.ac.jp/ m-mat/MT/emt.html*, 8 Apr 2008.
[125] R. Yaffee and M. McGee, *Introduction to Time Series Analysis and Forecasting*. Academic Press, first ed., 2000.
[126] G. Box and G. Jenkins, *Time Series Analysis Forecasting and Control*. Holden Day, second ed., 1976.
[127] E. Weisstein, "Modified bessel function of the second kind," *http://mathworld.wolfram.com/-ModifiedBesselFunctionoftheSecondKind.html*, 12 Mar 2008.
[128] National Institute of Standards and Technology (NIST), "Federal information processing standards (FIPS) publication 197," pp. 1–51, Nov 2001.
[129] R. McEvoy, J. Curran, P. Cotter, and C. Murphy, "Fortuna: Cryptographically secure pseudo-random number generation in software and hardware," *Irish Signals and Systems Conference*, pp. 457–462, 28-30 June 2006.
[130] National Institute of Standards and Technology, "Batteries of statistical tests for random number generators," *http://csrc.nist.gov/groups/ST/toolkit/rng/batteries_stats_test.html*, 8 Feb 2009.
[131] B. Sklar, *Digital Communications: Fundamentals and Applications*. Prentice Hall, second ed., 2000.
[132] B. Weaver, "A new, high efficiency, digital, modulation technique for AM or SSB sound broadcasting applications," *IEEE Transactions on Broadcasting*, vol. 38, no. 1, pp. 38–42, Mar 1992.
[133] G. Sandhu and S. Berber, "Investigation on operations of a secure communication system based on the chaotic phase shift keying scheme," in *Proceedings of the Third International Conference on Information Technology and Applications*, pp. 584–587, IEEE Computer Society, 2005.
[134] M. Lee, Y. Hong, and K. Shore, "Experimental demonstration of VCSEL-based chaotic optical communications," *IEEE Photonics Technology Letters*, vol. 16, pp. 2392–2394, Oct. 2004.
[135] J. McClellan and T. Parks, "A personal history of the Parks-McClellan algorithm," *Signal Processing Magazine, IEEE*, vol. 22, no. 2, pp. 82–86, March 2005.
[136] R. Crochiere and L. Rabiner, "Interpolation and decimation of digital signals: a tutorial review," *Proceedings of the IEEE*, vol. 69, pp. 300–331, Mar 1981.
[137] D. Chester, "Multirate filtering lunch&learns," *Harris Corporation*, Sessions 1-5, Feb 2007.
[138] Maxim IC Corporation, "MAX5895 datasheet: 16-bit 500 Msps interpolating and modulating dual DAC with CMOS inputs," *http://datasheets.maxim-ic.com/en/ds/MAX5895.pdf*, pp. 1–32, 8 Aug 2007.
[139] Cross Technologies, "2015-25 upconverter, 70 MHz to 2.0-2.5 GHz," *http://www.crosstechnologies.com/data_sheets/2015-25_DATA_SHEET.pdf*, p. 1, 8 Aug 2007.
[140] Cross Technologies, "Instruction manual: Model 2015-25 upconverter," *http://www.crosstechnologies.com/manuals/2015-25_MANUAL.pdf*, pp. 1–16, 8 Aug 2007.
[141] Tektronix, "TLA 700 Factsheet," *http://www.tek.com/Measurement/cgi-bin/framed.pl?Document=/-Measurement/Products/factsheet/tla700/*, 8 Aug 2007.
[142] Hewlett Packard (Agilent), "Installation and verification manual: HP8566B spectrum analyzer," pp. 1–115, 1993.
[143] D. He, L. Jiang, H. Zhu, and G. Hu, "Phase tracking of CDMA spreading sequences using dynamic chaotic synchronization," *ISCAS*, vol. 4, pp. 282–285, May 2001.
[144] Cross Technologies, "2016-25 downconverter, 2.0-2.5 GHz to 70 MHz," *http://www.crosstechnologies.com/data_sheets/2016-25_DATA_SHEET.pdf*, p. 1, 8 Aug 2007.
[145] Cross Technologies, "Instruction manual: Model 2016-25 downconverter," *http://www.crosstechnologies.com/manuals/2016-25_MANUAL.pdf*, pp. 1–16, 8 Aug 2007.
[146] M. Irsigler and B. Eissfeller, "Comparison of multipath mitigation techniques with consideration of future signal structures," *Proceedings of International Technical Meeting of the Satellite Division of the Institute of Navigation*, pp. 2584–2592, Sep 9-12 2003.
[147] G. Mcgraw and M. Braasch, "Mitigation using gated and high resolution correlator concept," *Proceedings of the National Technical Meeting of the Satellite Division of the Institute of Navigation*, pp. 333–342, Jan. 25-27 1999.
[148] B. Parkinson and S. J., "Global Positioning System: Theory and Practice, Volumes I and II," *American Institute of Aeronautics and Astronautics*, 1996.
[149] S. Townsend and P. Fenton, "A practical approach to the reduction of pseudorange multipath errors in a L1 GPS receiver," *ION-GPS 94*, pp. 143–148, Sep 20-23 1994.
[150] F. Harris, "Performance and design of Farrow filter used for arbitrary resampling," *13th International Conference on DSP Proceedings*, vol. 2, pp. 595–599, July 1997.
[151] C. Barrow, "A continuously variable digital delay element," *ICAS-88*, vol. 3, pp. 2641–2645, June 1988.

[152] F. Gardner, "Interpolation in digital modems part-I: Fundamentals," *IEEE Transactions on Communications*, vol. 41, pp. 502–508, March 1993.
[153] M. Sangriotis and I. Xezonakis, "Digital Costas loop-like PLL for the carrier recovery of a QPSK signal," *Electronics Letters*, vol. 29, pp. 897–899, May 1993.
[154] D. Boritzki, "Basic design of phase locked loops," *Harris Corporation*, pp. 1–80, May 2007.
[155] J. Brand, "Practical on-the-move satellite communications for present and future mobile warfighters," *Military Communications Conference*, pp. 625–629 Vol. 1, Oct. 2005.
[156] D. Chester and A. Michaels, *Adaptive Correlation*. Harris Corporation patent (pending) GCSD-1952, 2008.
[157] D. Chester and A. Michaels, *Featureless Coherent Chaotic Amplitude Modulation*. Harris Corporation patent (pending) GCSD-1993, 2008.
[158] A. Michaels, *Selective Noise Cancellation of a Spread Spectrum Signal*. Harris Corporation patent (pending) GCSD-2022, 2008.
[159] D. Chester and A. Michaels, *Spread Spectrum Communications System and Method Utilizing Chaotic Sequence*. Harris Corporation patent (pending) GCSD-1874, 2007.
[160] D. Chester and A. Michaels, *Chaotic Spread Spectrum Communications System Receiver*. Harris Corporation patent (pending) GCSD-1990, 2008.
[161] Wolfram, "Wolfram Mathematica online integrator," *http://integrals.wolfram.com/index.html*, 12 Feb 2009.
[162] J. Spilker and D. Magill, "The delay lock loop discriminator: an optimum tracking device," *Proceedings IRE*, vol. 49, pp. 1403–1416, Sep 1961.
[163] J. Betz and K. Kolodziejski, "Extended theory of early-late code tracking for a bandlimited GPS receiver," *Journal of the Institute of Navigation*, vol. 47, pp. 211–226, Fall 2000.
[164] C. Hegarty, E. Power, and B. Fonville, "Accounting for timing biases between GPS, modernized GPS, and Galileo signals," 36^{th} *Annual Precise Time and Time Interval (PTTI) Meeting*, pp. 307–317, Dec 2004.
[165] A. Michaels and D. Chester, *Normalization Techniques for Non-stationary Chaotic Communications Waveforms*. Harris Corporation patent (pending) GCSD-2068, 2008.
[166] J. M. Hedenberg, "Characterization of binary offset carrier (BOC) systems coexisting with other wideband signals," *Air Force Institute of Technology, Wright Patterson AFB*, pp. 1–74, Dec 2005.
[167] G. Longo, "Communication in the presence of jamming - an information theoretic approach," *Secure Digital Communications*, no. 279, pp. 127–167, 1983.
[168] R. McEliece, "The jamming game," *Research Trends in Military Communications*, no. CSI-83-12-1, pp. 116–123, 1983.
[169] A. Michaels and D. Chester, *RAKE Receiver for Chaotic Communications*. Harris Corporation patent (pending) GCSD-2067, 2008.
[170] Y. Wen, W. Huang, and Z. Zhang, "CAZAC sequence and its application in LTE random access," *IEEE Information Theory Workshop, Chengdu*, pp. 544–547, Oct 2006.
[171] A. Kebo, I. Konstantinidis, J. Dellomo, and J. Sieracki, "Ambiguity and sidelobe behavior of CAZAC coded waveforms," *2007 IEEE Radio Conference*, pp. 99–103, Apr 2007.
[172] J. Benedetto and J. Donatelli, "Ambiguity function and frame-theoretic properties of periodic zero-autocorrelation waveforms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 1, pp. 6–20, June 2007.
[173] A. Michaels and D. Chester, *CAZAC-based Anti-Jam Waveform*. Harris Corporation patent (pending) GCSD-2055, 2008.
[174] A. Michaels and D. Chester, *Chaotic AJ Waveform in HF Extended Line-of-Sight Applications*. Harris Corporation patent (pending) GCSD-2056, 2008.
[175] M. Pursley, "Performance evaluation for phase-coded spread-spectrum multiple-access communication: System analysis," *IEEE Transactions on Communications*, vol. 25, pp. 795–799, Aug 1977.
[176] A. Salmasi and K. Gilhousen, "On the system design aspects of code division multiple access (CDMA) applied to digital cellular and personal communications networks," *41st IEEE Vehicular Technology Conference*, pp. 57–62, May 1991.
[177] N. Kong and L. Milstein, "Average SNR of a generalized diversity selection combining scheme," *IEEE Communications Letters*, vol. 3, pp. 57–59, Mar 1999.
[178] Z. Zvonar and D. Brady, "Multiuser detection in single-path fading channels," *IEEE Transactions on Communications*, vol. 42, pp. 1729–1739, Feb/Mar/Apr 1994.
[179] R. Lupas and S. Verdu, "Near-far resistance of multiuser detectors in asynchronous channels," *IEEE Transactions on Communications*, vol. 38, pp. 496–508, Apr 1990.
[180] G. Mazzini, G. Setti, and R. Rovatti, "Chaotic complex spreading sequences for asynchronous DS-CDMA: System modeling and results," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, pp. 937–947, Oct 1997.
[181] T. Yang and L. Chua, "Chaotic digital code-division multiple access (CDMA) communication systems," *Intl. Journal of Bifurcation and Chaos*, vol. 7, pp. 2789–2805, Dec 1997.

[182] V. Milanovic and M. Zaghloul, "Improved masking algorithm for chaotic communications systems," *Electronics Letters*, vol. 32, pp. 11–12, Jan 1996.
[183] K. Deergha Rao and B. Raju, "Improved robust multiuser detection in flat fading synchronous non-Gaussian channels using chaotic spreading," *15th International Conference on Digital Signal Processing*, pp. 371–374, July 2007.
[184] D. He and H. Leung, "Quasi-orthogonal chaotic CDMA multi-user detection using optimal chaos synchronization," *IEEE Transactions on Circuits and Systems II*, vol. 52, pp. 739–743, Nov. 2005.
[185] Y.-S. Lau, J. Jusak, and Z. Hussain, "Blind adaptive multiuser detection for chaos CDMA communication," *TENCON 2005, IEEE Region 10*, pp. 1–5, Nov. 2005.
[186] J. Yu, H. Li, Y.-D. Yao, and N. Vallestero, "LPI and BER performance of a chaotic CDMA system," *IEEE 64th Vehicular Technology Conference*, pp. 1–5, Sept. 2006.
[187] R. Rovatti, G. Mazzini, and G. Setti, "Enhanced rake receivers for chaos-based DS-CDMA," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, pp. 818–829, Jul 2001.
[188] A. Michaels and D. Chester, *Chaotic Multiple Access Communications: Shared Sequences, Coordinated Sequence Offsets*. Harris Corporation patent (pending) GCSD-2060, 2008.
[189] A. Michaels and D. Chester, *Chaotic Multiple Access Communications: Independent Orthogonal Sequences*. Harris Corporation patent (pending) GCSD-2061, 2008.
[190] A. Michaels and D. Chester, *Chaotic Multiple Access Communications: Coordinated Orthogonal Sequences*. Harris Corporation patent (pending) GCSD-2062, 2008.
[191] T. Sathyan and T. Kirubarajan, "Secure communication using chaotic systems and Markovian jump systems," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1932–1937 vol.2, Oct. 2003.
[192] A. Michaels and D. Chester, *Protected Chaos-based Multiple Access Communications: Additive Gaussian Masking*. Harris Corporation patent (pending) GCSD-2063, 2008.
[193] A. Michaels and D. Chester, *Protected Chaos-based Multiple Access Communications: Stationary and Non-stationary Phase Rotations*. Harris Corporation patent (pending) GCSD-2064, 2008.
[194] A. Michaels and D. Chester, *Protected Chaos-based Multiple Access Communications: Time-Division Sequence Modulation*. Harris Corporation patent (pending) GCSD-2065, 2008.
[195] A. Michaels, *Multi-Tier Adhoc Networking Neighbor Discovery Techniques*. Harris Corporation patent (pending) GCSD-2071, 2008.
[196] A. Michaels, *Multi-Tier DTDMA Communications*. Harris Corporation patent (pending) GCSD-2072, 2008.
[197] A. Michaels and D. Chester, *Symbol Duration Dithering for Secured Chaotic Communications*. Harris Corporation patent (pending) GCSD-2069, 2008.
[198] D. Chester and A. Michaels, *Continuous Time Chaos Dithering*. Harris Corporation patent (pending) GCSD-2058, 2008.
[199] D. Chester and A. Michaels, *Discrete Time Chaos Dithering*. Harris Corporation patent (pending) GCSD-2059, 2008.
[200] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge Univ. Press, second ed., 1999.
[201] A. Michaels and D. Chester, *Cryptographic System Incorporating a Digitally Generated Chaotic Numerical Sequence*. Harris Corporation patent (pending) GCSD-2017, 2008.
[202] A. Michaels and D. Chester, *High-Speed Chaos-based Cryptographic System*. Harris Corporation patent (pending) GCSD-2066, 2008.
[203] A. Michaels and D. Chester, *Constant Energy Symbols in Chaotic Communications for Optimal BER Performance*. Harris Corporation patent (pending) GCSD-2070, 2008.

Vita

Alan J. Michaels is a Systems Engineer at the Harris Corporation, where he has worked on a variety of defense communications projects since 6/2005, including Naval sensor systems, ad-hoc networking, GPS ground equipment, and numerous basic/applied research areas, resulting in 35 pending U.S. patents and 1 R.O.C. pending patent. Alan has earned a total of seven college degrees and been involved in a number of extracurricular activities while at Georgia Tech. Technology innovation, R&D, and strategic technology management are particular interests.

Education:

Georgia Institute of Technology:

- ▷ 9/1998-12/2000: B.S. in Electrical Engineering (highest honors)
- ▷ 1/2001-5/2001: M.S. in Electrical and Computer Engineering
- ▷ 5/2001-5/2003: Certificate in Social and Personality Psychology
- ▷ 5/2001-5/2003: B.S. in Applied Mathematics (highest honors)
- ▷ 5/2001-5/2003: M.S. in Applied Mathematics
- ▷ 5/2003-5/2005: M.S.in Operations Research
- ▷ 8/2006-8/2009: Ph.D. in Electrical and Computer Engineering

Carnegie Mellon, Tepper School of Business

- ▷ 8/2005-5/2008: M.B.A. with specializations in Strategy and Finance

Distinctions:

- ▷ Georgia HOPE, Governor's, GA Merit and Tandy Technology scholar, 1998-2000
- ▷ Texas Instruments/Georgia Tech Analog Consortium Fellow, 2001
- ▷ Instructor, GT Study Abroad program at Oxford Univ, 2002
- ▷ Georgia Tech Graduate Student Body President, 2002-2003
- ▷ National Academy of Engineering's "Engineer of 2020" Participant, 2002
- ▷ Georgia Tech School of ECE Outstanding Graduate Teaching Assistant, 2002&2004
- ▷ Georgia Tech $\pi\mu\epsilon$ President and HSMC creator, 2003-2005
- ▷ Instructor, GT Psych 1000, 2004
- ▷ National Science Foundation's East Asian Summer Institute Fellow, 2004
- ▷ MacArthur Foundation's Sam Nunn International Security Fellow, 2004-2005